VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Identifikace uživatelů sociálních sítí a digitálních knihoven

Social Network and Digital Library User's Identification

2014 Bc. Adam Ondrejka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Adam Ondrejka**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Identifikace uživatelů sociálních sítí a digitálních knihoven
Social Network and Digital Library User's Identification

Zásady pro vypracování:

Uživatelé sociálních sítí ne vždy zveřejní svou celou identitu, jejich aktivity zpravidla přesahují jednu sociální síť. Akademičtí uživatelé se často sdružují v uživatelských skupinách, které jim umožňují diskutovat například o konferenci, knize… v takových skupinách, či specializovaných sociálních sítích, již o sobě zveřejní informací více. Akademičtí pracovníci rovněž publikují výsledky svého výzkumu. Publikace bývají dostupné prostřednictvím digitálních knihoven, které jako přidanou hodnotu poskytují metadata. Dílčím úkolem projektu je hledat v sociálních sítích vytipované skupiny uživatelů, nalezení jejich vazeb na další uživatele, zájmovou doménu a podobně prohledání digitálních knihoven s cílem ztotožnit uživatele z různých sociálních sítích, nebo vystupujících jako autoři v digitálních knihovnách.

1. Seznamte se se stavem poznání v oblasti sociálních sítí a digitálních knihoven.
2. Analyzujte možnosti pro identifikaci uživatele v různých prostředích sociálních sítí a digitálních knihoven.
3. Vyberte vhodné postupy pro (polo)automatickou identifikaci uživatele a určení jeho odborných zájmů.
4. Ve zvolených sociálních sítích analyzujte vybranou skupinu výzkumníků, hledejte jejich propojení a zájmy. Experiment předem konzultujte s vedoucím.
5. Pokuste se doporučit vybraný odborný text (doplněný o metadata) skupině výzkumníků.
6. Analyzujte postup a dosažené výsledky.

Seznam doporučené odborné literatury:

[1] Bing Liu. Web DataMining; Exploring Hyperlinks, Contents, and Usage Data. Chapter Link Analysis, pages 237-272. Corrected 2nd printing 2008 ISBN-10 3-540-37881-2 Springer Berlin Heidelberg New York, 2007
[2] A. K. Jain, M. N. Murty, P. J. Flynn. Data clustering: a review. ACM Computing Surveys (CSUR), Volume 31 , Issue 3 (September 1999), ISSN:0360-0300, Pages: 264 - 323, ACM 1999
[3] I-Chin Wu, Che-Ying Wu. Using internal link and social network analysis to support searches in Wikipedia: A model and its evaluation. April 2011, Journal of Information Science , Volume 37 Issue 2. ACM DL
[4] José Luis Ortega, Isidro F. Aguillo. Visualization of the Nordic academic web: Link analysis using social network tools. July 2008, Information Processing and Management: an International Journal , Volume 44 Issue 4
[5] Ivan Herman, Guy Melançon, M. Scott Marshall. Graph Visualization and Navigation in Information Visualization: A Survey January 2000, IEEE Transactions on Visualization and Computer Graphics , Volume 6 Issue 1
Publisher: IEEE Educational Activities Department

[6] Cameron Marlow, Mor Naaman, Danah Boyd, Marc Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. August 2006, HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia, ACM DL
Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. August 2006, KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining
[7] Nástroj RaVis. Online http://code.google.com/p/birdeye/wiki/RaVis
[8] Albert-László Barabási: V pavučině sítí, Paseka, edice Fénix,2005, váz. 280 str., ISBN 80-7185-751-3.
[9] Popis API Facebooku pro vývojáře, http://developers.facebook.com/

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. RNDr. Petr Šaloun, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA 1

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 6. května 2014 .....

Rád bych na tomto místě poděkoval především svému vedoucímu diplomové práce panu doc. RNDr. Petru Šalounovi, Ph.D. za odborné vedení, cenné rady a trpělivost při tvorbě této práce.

Abstrakt

Tato diplomová práce je zaměřena na identifikaci uživatelů sociálních sítí a digitálních knihoven a na doporučování publikací vědeckým výzkumníkům. Toho je dosaženo pomocí analýzy veřejně dostupných dat a publikacích z digitálních knihoven a informací zveřejněných na sociálních sítích. Obsahuje původní návrh a implementaci obou algoritmů a navržený postup je ověřen na datech ze dvou konferencí. Správným odhadem domény, ve které autor publikuje, a vhodně zvolenými klíčovými slovy a spoluautory poměrně efektivně nalezneme identity uživatelů a doporučíme odpovídající publikace z oblasti jejich výzkumu.

Klíčová slova: identifikace uživatele, sociální sít, digitální knihovna, doporučení publikace, vytěžování dat/data mining, webové inženýrství

Abstract

This master thesis is focused on the identification of users of social networks and digital libraries and recommending scientific publications to researchers. This is accomplished by analisis of publicly available metadata about authors and publications from digital libraries and information posted on social networks. This work contains original design and implementation of both algorithms and proposed approach is tested using data from two conferences. The correct estimation of domain in which is author publishing, well-chosen keywords and co-authors quite effectively finds the identity of users and recommend appropriate publications in the field of research.

Keywords: identify user, social network, digital library, data mining, recommend publication, web engineering

Seznam použitých zkratek a symbolů

ACM	–	Association for Computing Machinery
API	–	Application Programming Interface
CSS	–	Cascading Style Sheets
DNA	–	Deoxyribonucleic Acid
HTML	–	HyperText Markup Language
HTTP	–	HyperText Transfer Protocol
IEEE	–	Institute of Electrical and Electronics Engineers
IP	–	Internet Protocol
JSON	–	JavaScript Object Notation
LDA	–	Latent Dirlecht Allocation
MVC	–	Model-View-Controller
NLP	–	Natural Language Processing
NLTK	–	Natural Language Toolkit
OLAP	–	Online Analytical Processing
ORM	–	Object-Relation Mapping
REST	–	Representational State Transfer
RPC	–	Remote Procedural Call
SOAP	–	Simple Object Access Protocol
SQL	–	Structured Query Language
URL	–	Uniform Resource Locator
XML	–	Extensible Markup Language

Obsah

1 Úvod **5**

2 Identifikace uživatelů – stav poznání **7**

3 Digitální knihovny a sociální sítě **9**
3.1 Vybrané digitální knihovny 9
3.2 Vybrané sociální sítě 10

4 Analýza problému **14**
4.1 Identita uživatelů 14
4.2 Hledání podobnosti 16
4.3 Částečná shoda textových řetězců 20
4.4 Základní tvary klíčových slov 21

5 Hledání identit uživatelů na sociálních sítích a digitálních knihovnách **23**
5.1 Komunikace s vnějším světem 23
5.2 Zpracování publikací 24
5.3 Hledání autora v digitálních knihovnách 27
5.4 Hledání identit na sociálních sítích 30
5.5 Doporučování publikací 35
5.6 Použité technologie 38

6 Výsledky experimentu **41**
6.1 Experiment hledání identit 41
6.2 Experiment doporučování publikací 41

7 Závěr **44**

8 Reference **46**

Přílohy **47**

A Obsah přiloženého CD a článek HyperText 2014 **48**

Seznam tabulek

1 Slova s pravděpodobností výskytu v dokumentu 19
2 Výsledky experimentu hledání publikací 42
3 Výsledky experimentu hledáni identit 42
4 Výsledky experimentu doporučení publikací 42

Seznam obrázků

1 Domovská stránka knihovny ACM Digital Library 10
2 Domovská stránka knihovny IEEExplore 11
3 Domovská stránka sítě ResearchGate po přihlášení 12
4 Mapování atributů na profilech sítí Facebook a Google+ 15
5 Komunikace s vnějším světem 24
6 Návrh algoritmu zpracování publikace 25
7 Nalezené skupiny autorů pro vstup *John Doe*. Skupiny jsou vytvořeny podle spoluautorů a porovnáním klíčových slov publikací. 30
8 Profil uživatele sítě LinkedIn 33
9 Veřejný profil uživatele na síti ResearchGate 34
10 Grafy funkcí časových dynamik 37
11 Třídní diagram aplikace 40

Seznam výpisů zdrojového kódu

1 Pseudokód nalezení dodatečných klíčových slov 26
2 Pseudokód nalezení identity uživatele na digitálních knihovnách 28
3 Pseudokód nalezení identity uživatele na LinkedInu 31
4 Pseudokód nalezení identity uživatele na ResearchGate 33
5 Pseudokód doporučení publikací 36

1 Úvod

Velký rozmach sociálních sítí a prvků na internetu zapříčinil problém v podobě velké fragmentace účtů způsobuje obtížnou práci s uživatelem jako s jedinečnou bytostí. Existují sice projekty identifikující uživatele pomocí jediného účtu (např. OpenID, MojeID,...), ale nasazení těchto služeb se sociálním sítím zpravidla vyhýbá.

Hlavním cílem práce je hledání výzkumníků napříč vybranými sociálními sítěmi na základě jejich publikací a případných zájmů zveřejněných na profilech. Druhým úkolem je navrhnout a implementovat doporučovací algoritmus a systém založený na analýze sesbíraných dat o výzkumnících. Uživatel zadá základní informace o článku a algoritmus automaticky vyhledá nejvíce odpovídající výzkumníky dle jejich publikační činnosti. Tento algoritmus poté může najít využití například pro hledání vhodných oponentů bakalářských, diplomových a disertačních prací, případně může být součástí komplexnější aplikace pro pořádání konferencí. Přihlášeným článkům mohou být na jeho základu jednodušeji přidělení vhodní hodnotitelé na základě jejich předchozí publikační činnosti. Hledáním podobnosti publikací pouze dle klasického a běžně používaného přístupu porovnáváním slov nemusí být pro náš úkol příliš vhodné. Máme k dispozici malé množství informací a pouhým porovnáváním slov textu bychom se brzy dopustili informačního šumu. Bude zřejmě nezbytné pokusit se porozumět textu alespoň jednoduchým způsobem a hledat k vytipovaným slovům jejich synonyma. Pro tyto účely využijeme volně dostupný slovník WordNet.

Druhá kapitola shrnuje aktuální stav v oblasti výzkumu problematiky identity uživatelů, doporučování odborných článků a hledáním podobností mezi publikacemi.

Třetí kapitola stručně popisuje výhody a omezení digitálních knihoven a sociálních sítí. Také podrobněji rozebírá vybrané internetové služby z každé kategorie.

Další část textu se jíž zabývá samotnou analýzou problematiky a detailněji uvádí možné přístupy k řešení na teoretické úrovní. Čtenář se zde dočte o principech hledání identit uživatelů, dozví se o modelech a metodách hledání podobnosti mezi dokumenty v Booleovském modelu, vektorovém prostoru a pomocí shlukové analýzy. Následují algoritmy částečné shody textových řetězců (tzv. fuzzy match) pro porovnávání mírně odlišných textů. V kapitole je rovněž zmíněna tvorba kořene slov a získání základních tvarů, které jsou důležitým procesem pro další postup nalezení identity uživatele.

Následovat bude konkrétní návrh výše zmíněných algoritmů. Nejprve se čtenář informuje o možnostech komunikace s vybranými digitálními knihovnami a sociálními sítěmi. kapitola pokračuje částí o zpracování publikace, normalizaci slov a získání dodatečných klíčových slov z textu abstraktu sloužící k určení domény autora. Dále je popsán samotný algoritmus hledání identity uživatele, zvlášť pro digitální knihovny a dvě sociální sítě včetně algoritmu pro doporučování publikací uživatelům.

V další kapitole je stručně popsána implementace experimentálních algoritmů, včetně návrhu databáze, komunikace v systému a popise použitých technologií a knihoven.

Předposlední kapitola ověřuje funkčnost navržených algoritmů. Problém hledání identity uživatele na digitálních knihovnách a sociálních sítích je otestován na 30-ti uživatelích. Ověření doporučování publikaci bylo aplikováno na případovou studii pořádání

konferencí. Pro přihlášené články jsou hledání nejvhodnější hodnotitelé, experiment byl prověřen na výstupech dvou reálných konferencí, lokální a globální. Controller Závěrečná kapitola shrnuje dosažené výsledky a naznačuje možný směr dalšího vývoje a výzkumů na základě získaného poznání při práci na této diplomové práci.

2 Identifikace uživatelů – stav poznání

Sociální sítě si díky své vysoké popularitě v poslední době vysloužily pozornost u vědců a staly se terčem mnoha výzkumů a experimentů. Cíl této diplomové práce, hledání identity uživatelů sociálních sítí a digitálních knihoven, ovšem dosud unikal širšímu zájmu a výzkumu soudě dle vydaných publikací přímo na toto téma. Analýzou problematiky jako celku a návrhem algoritmu, obojí je podrobně vysvětleno v následujících kapitolách, dostaneme více dílčích problémů s již dostatečným zájmem vědců i firem, s dostatečně publikovanými výsledky pokusů.

Kniha *Data Mining: Concepts and Techniques*[1] se detailně věnuje popisu problémů a technik souvisejících s dolováním dat z webu. Čtenář se dočte o možnostech zpracování textu na webu, ukládání velkého množství dat do databázových modelů jako jsou OLAP a DataCube, o zjišťování a analyzování sociálních sítí a vztahů mezi získanými atributy.

V projektu RefferalWeb[2] vedeném Kautzem byla poprvé vyzkoušena technika web miningu pro extrakci sociální sítě. Cílem bylo vytvořit nástroj pro tzv. *zřetězené doporučení* (angl. refferal chaining), kdy se hledají odborníci s danou odborností blízkých k uživateli systému. Příkladem otázky, která může být systému položena je „nalezni všechny odborníky na simulované žíhání, kteří jsou nejdále 3 uzly v síti ode mne“.

Jan Vosecky ve svém článku *User identification accross multiple social networks* [3] popisuje možný postup vhodný k identifikaci uživatele napříč různých sociálních sítích založený na porovnávání profilů ve vektorovém prostoru. Na profilech sítí Facebook a StudiVZ vybírá vhodná klíčová slova a mezi nimi hledá podobnost. Využívá také tzv. *přibližné shody (angl. fuzzy matching)* a to z důvodů rozdílného vyjádření stejného objektu. Například jméno „Jan Vosecky“ je na jiném profilu napsáno jako „Vosecky Jan“ nebo „J. Vosecky“, pouhým porovnáním přesné shody bychom tedy získali nepřesný výsledek.

V publikaci *User Profile Matching in Social Networks* [4] využívá Elie Raad podobných principů ve vektorovém principů. Přidává navíc váhu jednotlivým klíčovým slovům a mimo ně porovnává také celé věty na profilech uživatelů. Výsledek, zda se uživatelé shodují nebo ne, je nakonec rozhodnut na základě skóre podobnosti mezi profily a ručně nastavenou hodnotou.

Z oblasti digitálních knihoven publikoval například Martin Germán v článku *Finding similar research papers using language models* [5] metody k nalezení podobných vědeckých článku pomocí jazykového modelu. Princip je založen na odhadu, zda jeden text dokáže vygenerovat slova z textu druhého. Čím je těchto slov více, tím větší je samozřejmě podobnost. Odhad je založen na předchozí analýze abstraktů, klíčových slov a následovného strojového učení a může být alternativou, případně doplněním k porovnávání publikací ve vektorovém prostoru.

Mutschke a Haaseová v článku *Collaboration and Cognitive Structures in Social Science Research Fields* provedli síťovou analýzu uživatelů na základě bybliografických záznamech obsahující klíčová slova publikací. Nejprve seskupili klíčová slova publikací do témat podle spoluautorů a na připravená data aplikovali analýzu sociálně-kognitivní sítě.

Druhý cíl diplomové práce - doporučení publikací jiným autorům můžeme zařadit do problému doporučování obecně. Tato oblast je důležitá a řešena zejména v e-komerční sféře. Firmy (např. elektronické obchody) chtějí nakupujícím nabídnout další adekvátní

zboží s co nejvyšší pravděpodobností, že jej přikoupí k již vybranému zboží. Toho docílí zejména kolektivní inteligencí, která je založena na analýze nákupních návyků jednotlivých uživatelů a hledání podobností s jinými. Postupy jsou popsány v knize *Programming collective inteligence* [6] a opět využívají převodu entit do vektorového prostoru a hledání vzdálenosti mezi nimi. S podobnými postupy se mimo jiné můžeme setkat i při doporučování vhodných filmů na základě hodnocení jiných, doporučení hudby apod.

3 Digitální knihovny a sociální sítě

Digitální knihovny jsou bez debaty velkým přínosem vědě a školství. V moderním informačním věku znamenají hlavně téměř okamžitý přístup k novým informacím a poznáním a odstraňují tak jednu z nevýhod tradičních knihoven a sice dostupnosti publikací a rychlosti zveřejnění nových poznatků. Na druhou stranu ovšem objevuje jiná nevýhoda v podobě velké roztříštění autorů v různých digitálních knihovnách a tím obtížnějšího hledání identity konkrétního autora.

Sociální sítě, jak je již z jejich označení patrné, slouží především ke spojování a tím i vzniklou výměnu informací mezi uživateli a společností v rámci propojených společenských kruhů. Kvůli specifickému zaměření jednotlivých služeb nebývá aktivita uživatelů omezena pouze na jednu sociální síť, ale přesahuje i do jiných. Obecně se dá říci, že právě data uživatelů jsou pro každou sociální síť tím nejcennějším, co mají, proto je téměř pravidlem, že své informace střeží a neposkytují k nim třetím stranám přístup.

3.1 Vybrané digitální knihovny

Výhodou digitálních knihoven, na rozdíl od sociálních sítí je, že umožňují přístup k základním údajům publikací, tzv. metadatům. Jde o základní informace popisující dílo, patří mezí ně: *titulek*, *autoři*, *klíčová slova*, *rok vydání*, *abstrakt*, *instituce* a *reference*. Podle těchto údajů dokáže člověk poměrně spolehlivě odhadnout doménu publikace, zda její autor je skutečně ten, koho hledal apod. Nevýhodou pak je, že každá digitální knihovna vrací tyto informace v trochu jiném formátu. Dostupné jsou příjmení autorů, ale ne již jejich jména. Instituce na kterých pracují nejsou v jednotném formátu, dokonce ani na stejné digitální knihovně. A některé služby navíc neposkytují autorova klíčová slova, což komplikuje další analýzu. Následuje stručná analýza vybraných potencionálních digitálních knihoven a sociálních sítích k návrhu algoritmu a následného ověření pomocí experimentu.

3.1.1 ACM Digital library

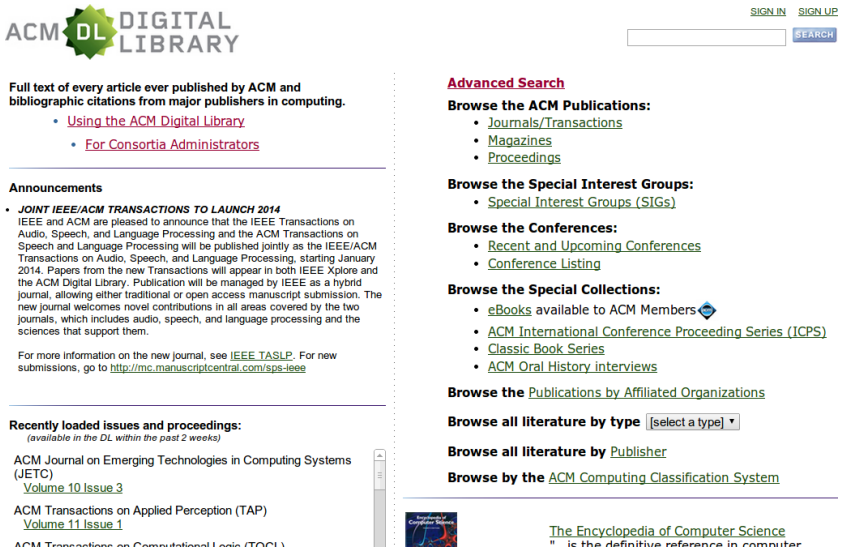
ACM Digital library [1] je digitální knihovna poskytující články, knihy a publikace především v oblasti počítačových věd. Obsahuje přes dva miliony příspěvků a řadí se tím mezi největší digitální knihovny vůbec. K získání metadat publikací bohužel nenabízí veřejné API, do strojové podoby je tedy potřeba získat rozborem zdrojového kódu HTML a vytvořením vlastního stahovacího robota.

Z metadat se ale na druhou stranu dají získat celá jména autorů a všechny ostatní potřebné informace.

3.1.2 IEEExplore

IEEEXplore [2] je služba organizace IEEE zaměřující se na tématiku technologií a inženýrství včetně počítačových věd. Čítá okolo tři a půl miliónů publikaci a rovněž se řadí mezí větší

[1] http://dl.acm.org
[2] http://ieeexplore.ieee.org/



Obrázek 1: Domovská stránka knihovny ACM Digital Library

digitální knihovny. Na rozdíl od předešlých stránek, nabízí veřejné API vracející výsledky v jednoduše strojově zpracovatelném formátu XML. Bohužel neuvádí celé jméno autorů, což budu muset zohlednit při dalším návrhu.

3.1.3 Springer Link

SpringerLink [3]je obecně zaměřená digitální knihovna německého původu. Obsahuje publikace technických, lékařských, právních a mnoha dalších oborů. Dle jimi zveřejněných informací obsahuje na osm milionů publikací. Po registraci je možno přistupovat k metadatům pomocí připraveného API, stejně jako v předchozím případě ale nezveřejňuje celá jména autorů. Navíc je omezeno počtem požadavků za vteřinu, při velkém zatížení dočasně zablokuje uživatele.

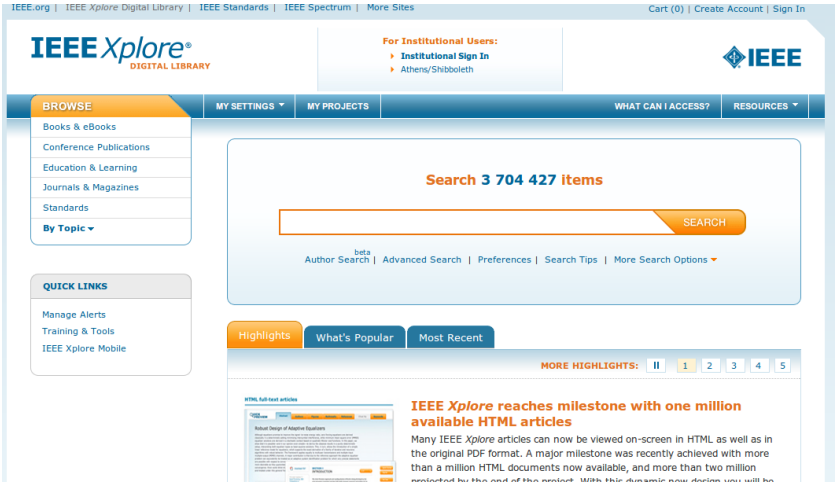
3.1.4 arXiv

ArXiv [4] je v porovnání s předchozími sítěmi menší knihovnou zaměřující se na technické obory jako jsou počítačové vědy, fyzika, matematika a finance. Databáze obsahuje necelý milion příspěvků, ale v ČR je známá a poměrně často využívána. Také neposkytuje celá jména autorů, ale to je vykompenzováno neomezeným veřejným API.

3.2 Vybrané sociální sítě

Sociální sítě neposkytují ve srovnání s digitálními knihovnami jednotné informace, dle kterých bych se dal přístup k nim sjednotit. Do jisté míry by se za ně dalo považovat jméno

[3]http://link.springer.com/
[4]http://arxiv.org/



Obrázek 2: Domovská stránka knihovny IEEExplore

autora a jeho spojení s jinými uživateli (přátele, sledující), často zde ale uživatelé volí přezdívky místo skutečných jmen, případně to sociální sítě dokonce vyžadují. Musíme tedy přistupovat ke každé webové službě individuálně a možnost propojení identity zvážit na základě dostupných dat na profilech a zaměření sítě.

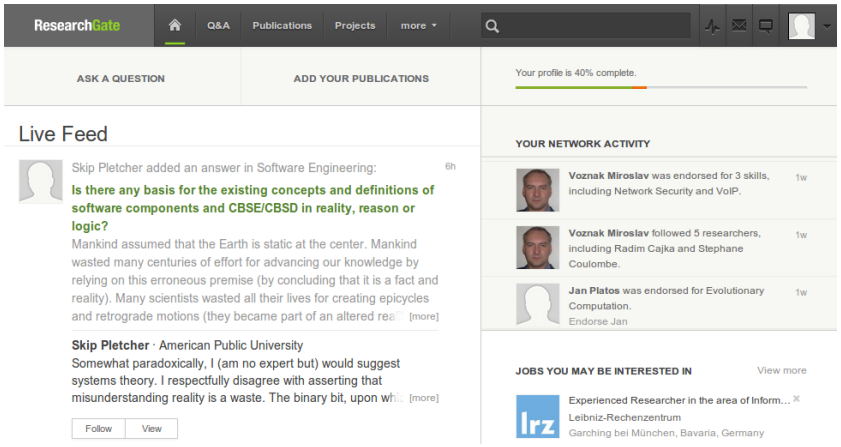
3.2.1 Facebook

Facebook[5] je největší a nejrozšířenější sociální sítí na světě s celkovým počtem aktivních uživatelů 1 300 000 (dle výzkumu StatisticBrain[6] roku 2014). Vznikl jako studentský projekt na americké univerzitě v roce 2005 a v následujících letech se masově rozšířil do všech koutů světa. Zaměření této sítě je především na komunikaci mezi přáteli, sdílení příspěvků a „lajkování" statusů jiných uživatelů. Na základech této kolektivní inteligence Facebook dále vyhodnocuje možné oblasti zájmů, doporučuje další příspěvky a zobrazuje cílenou reklamu. Na první pohled se zdá, že je ideálním zdrojem dat pro identifikaci uživatelů, což je na jednu stranu pravda, hlavně pro uvádění přátel, pracovních pozic a označení zajímavých stránek. Bohužel zavedením nového Graph API Facebook znemožnil jednoduché získání informací o přátelích a pro další návrh algoritmu je tedy síť nevhodná.

3.2.2 Google+

Google+ [7] je odpovědí Googlu na Facebook. Tato sociální síť byla spuštěna v roce 2011 a dodnes získala 540 milionů aktivních uživatelů. Stejně jako předešlá síť i tato je zaměřena na komunikaci mezi uživateli, kteří se seskupují do tzv. kruhů a je založena na sdílení

[5] http://www.facebook.com/
[6] http://www.statisticbrain.com/facebook-statistics/
[7] https://plus.google.com/



Obrázek 3: Domovská stránka sítě ResearchGate po přihlášení

stránek a publikace zážitků. Popularita bohužel nedosáhla úrovně Facebooku a Google+ se tak začal nuceně slučovat s ostatními službami Googlu. Bohužel není dostupné veřejné API, které by umožňovalo vyhledávání uživatelů v rámci sítě. Tento fakt se sice dá částečně obejít pomocí napojení na AJAXové volání v síti, i tak ale nejsme schopni získat dostatek informací o uživateli, pouze neúplný seznam jeho přátel.

3.2.3 LinkedIn

LinkedIn[8] je velká sociální síť zaměřena hlavně na profesní život uživatelů. Spuštěna byla v květnu 2003 a dnes se na ni měsíčně přihlašuje přes 260 milionů uživatelů. Jejich profily jsou veřejné (pozn. autora: v době psaní tohoto textu LinkedIn provedl změny a není možno ve veřejných profilech stránkovat a některé profily se staly neveřejnými) a obsahují zajímavé informace: *ostatní uživatele*, se kterými je ve spojení (mohli by být spoluautoři a kolegové), *pracovní pozice*, *schopnosti a dovednosti* a *vydané publikace*. LinkedIn po registraci umožňuje přistupovat k datům pomocí API, bohužel ale bez možnosti vyhledávání (dá se ovšem zažádat o výjimku).

3.2.4 ResearchGate

ResearcGate[9] je poměrně nová sociální síť se zaměřením na výzkumníky a jejich výzkum. jedná se v podstatě o digitální knihovnu s prvky sociálních sítích, autoři sledují ostatní autory a kolegy, prohlížejí si jejich publikace a přihlašují se do skupin se specifickým zaměřením, atd. Zajímavostí jistě je, že do ni zainvestoval i Bill Gates. Bohužel nenabízí veřejné API, přístup k datům bude muset být zajištěn webovým stahovačem ideálně s podporou JavaScriptu.

[8] https://www.linkedin.com
[9] http://www.researchgate.net/

3.2.5 Twitter

Pro úplnost dodávam Twitter[10], populární síť sloužící k výměně krátkých textových zpráv do 160 znaků. Oblíbena je zejména mezi celebritami (celosvětově), v ČR má naopak popularitu zejména mezi lidmi z IT oborů. Protože nabízí o uživatelích pouze informace v podobě krátkých statusů a jeho sledovatelů - followerů a to navíc ještě ve formě přezdívek, nebude do další analýzy zahrnuta.

3.2.6 Instagram

Instagram[11] pochází z nové generace sociálních sítí, která se zaměřuje na jedinou funkci, a to sdílení fotografií. Uživatelé sledují ostatní uživatele, hodnotí a komentují jejich nahrané fotografie. Opět je uvedena hlavně pro úplnost, kvůli nedostatku dalších informací není vhodná pro náš typ problému.

[10] https://twitter.com/
[11] http://instagram.com/

4 Analýza problému

Tato kapitola se zabývá vysvětlením problémů vedoucích k návrhu možného funkčního experimentálního algoritmu, který je cílem této práce. Shrnuje poznatky objevené při realizaci diplomové práce a naznačuje nástin možného řešení v teoretické rovině. Návrh algoritmu a konkrétní výběr metodik a postupů je popsán v následující kapitole.

4.1 Identita uživatelů

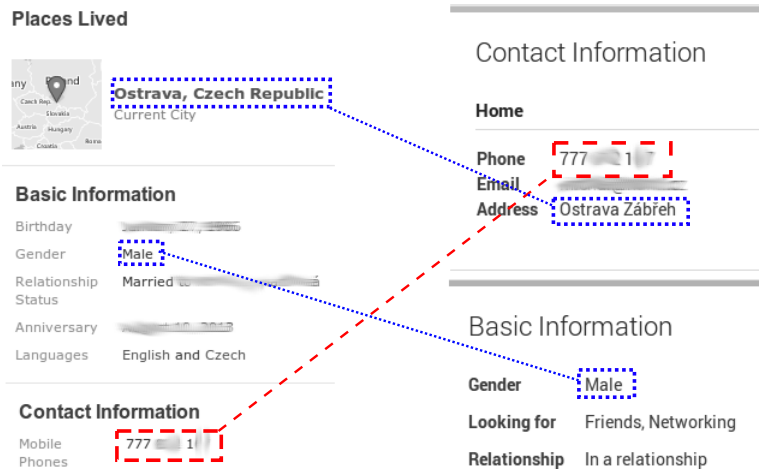
V ideálním světě má každý člověk své vlastní jedinečné jméno či jiný unikátní identifikátor, který jej jednoznačně určuje. Ve světe skutečném tomu tak bohužel není a mnoho autorů má podobné či naprosto stejná jména v příbuzné oblasti výzkumu nebo zcela odlišné. Představte si situaci, kdy se pokoušíte najít nějakého autora (např. člověka, kterého jste potkali na konferenci). Pokud zadáte jeho jméno do vyhledávače digitální knihovny, mohou nastat tyto situace:

- Autor má neobvyklé jméno nebo dokonce jedinečné. Nikdo jiný s tímto jménem neexistuje, najdeme jej tedy okamžitě.
- Autor má obvyklé jméno. Nalezneme mnoho lidí s tímto jménem, ale naštěstí zkoumají v různých oblastech, pracují na různých institucích nebo mají různé spolupracovníky. Dokážeme tedy poměrně snadno a přesně odhadnout osobu, kterou skutečně hledáme.
- Autor má velmi obvyklé jméno. Nalezneme spoustu stejně pojmenovaných lidí a bohužel někteří mají podobnou oblast výzkumu, stejné spolupracovníky nebo lidi pracující na stejné instituci.

Podobný scénář může nastat samozřejmě také při hledání uživatelů na sociálních sítích. Pro jednu osobu získáme více možných výsledků, pro člověka je tato komplikace částečně usnadněna fotografií, která je dostupná na většině podobných sítí. V případě automatizovaného strojového přiřazení tento fakt mnoho neřeší, alespoň ne triviálně. Jak je z výše popsaného patrné, poslední bod je velmi problematický dokonce i s ručním ověřením autora. Dále se tedy bude zaměřovat hlavně na první dva případy.

K dispozici máme data o uživateli, která sám poskytne a učiní veřejně dostupnými. Při zkoumání osobních profilů není obtížné postřehnout, že existuje množina určitých atributů, které uživatelé sdílejí napříč sociálními sítěmi. Ty se dají dále kategorizovat do dvou skupin, a sice *unikátní* a *obecné*. Na obrázku 4 je znázorněno mapování atributů mezí sociálními sítěmi Facebook a Google+. Červenou barvou jsou vyznačeny unikátní atributy, modrou obecné.

Mezi unikátní atributy můžeme zařadit například emailovou adresu a telefonní čísla. Tyto hodnoty jsou vysoce vypovídající o konkrétní identitě uživatele, bohužel ale jejich zveřejnění není tak časté, aby bylo možné spolehnout se na ně. Pokud již ovšem osoba na svých profilech uvede např. emailovou adresu a my ji dokážeme detekovat, v případě shody můžeme téměř jistě rozhodnout, že se jedná o tutéž osobu. Bohužel, opačný



Obrázek 4: Mapování atributů na profilech sítí Facebook a Google+

přístup není ekvivalentní, tedy pokud profily obsahují emailové adresy, ale neshodují se, nemůžeme rozhodnout o vztahu mezi profily. Uživatel totiž může používat více emailových adres a více telefonních čísel.

Při selhání komparace unikátních vlastností je vhodné přistoupit k porovnávání obecných atributů, patří mezi ně např. jméno, pohlaví, místo pobytu, přátelé atd. Samostatně jsou nevypovídající, porovnáváním jako celku jsme již schopni přesněji určit podobnost mezi uživateli. Pokud víme, které sítě budeme analyzovat a párovat, je vhodné předem provézt analýzu dostupných atributů a přidat jim jednotlivé váhy. V případě, že na osobních profilech nalezneme informaci o pohlaví a hodnota se liší, s jistotou vyloučíme spojení mezi identitami díky velké přidělené váze tomuto atributu. Přístup samozřejmě předpokládá vyplnění správných údajů na profilech.

V publikaci *User Profile Matching in Social networks* [4] je shoda profilů popsána ve dvou krocích:

1. *Nalezení prahové hodnoty*, podle které se posuzuje, zda profily patří pod jednu identitu nebo ne. Tato hodnota je vypočtena následovně:

$$th = f_{decision}(w(a_0), w(a_1, ..., w(a_n)) \tag{1}$$

kde

- th je prahová hodnota shody profilu
- $f_{decision}$ je rozhodující funkce
- a je zvolené atributy popisující uživatelský profil
- n je počet atributů
- w je váha přiřazená atributu

2. *Výpočet skóre podobnosti* mezi dvěma profily. Pro každou dvojici vybraných atributů se vypočítá podobnost dle vzorce:

$$sim'(P_1.a_i, P_2.a_i) = \frac{2 \times sim(P_1.a_i, P_2.a_i) \times w(a_i)}{1 + (sim(P_1.a_i, P_2.a_i) \times w(a_i)} \quad (2)$$

kde

- a_i je atribut použit k popisu profilu
- $P_1.a_i, P_2.a_i$ jsou dvě hodnoty atributu a_i na profilu P_1 a profilu P_2
- $w(a_i)$ je vypočtená/přiřazená váha k atributu $\in [0, 1]$
- $sim(P_1.a_i, P_2.a_i)$ je vypočtená podobnost mezi hodnotami atributu a_i v P_1 a P_2 $\in [0, 1]$
- $sim'(P_1.a_i, P_2.a_i)$ je nová vypočtená podobnost mezi hodnotami atributu a_i v P_1 a $P_2 \in [0, 1]$

Podobnost všech atributů se nakonec vyhodnotí rozhodující funkcí a pokud je výsledek vyšší než prahová hodnota, profily jsou podobné, tedy patřící jedné osobě.

4.2 Hledání podobnosti

V předchozí podkapitole byl zmíněn termín *podobnost*. Následující část textu bude věnována možnostem hledání podobnosti klíčových slov, dokumentů a atributů.

4.2.1 Booleovský model

Klasickým a jedním z prvních hojně nasazovaných modelem pro vyhledávání dotazů a podobností dokumentů je Booleovský model[7]. Zakládá na Booleovské logice obecné teorii množin, ve které jsou dokumenty reprezentovány jako množina termů. Výsledkem podobnosti, případně dotazu, jsou hodnoty 0 nebo 1. Mějme dokumenty definovány jako $D_i = \{t_{i,1}, t_{i,2}, ..., t_{i,n}\}$, kde

- t je term dokumentu D_i
- n je počet termů v jednotlivých dokumentech

Na dotaz $Q = (t_1 OR t_3) AND t_4$, získáme výsledky dokumentů, které obsahují termy t_1 a t_4 nebo t_3 a t_4.

Je tedy patrné, že přístup je založen na přesné shodě, dokument buď předepsanou podmínku - dotaz splňuje, nebo ne, neexistuje zde žádná míra podobnosti. Booleovský model rovněž nepočítá s ohodnocením důležitých slov, všechny termy mají stejnou váhu.

4.2.2 Model vektorového prostoru

Nejrozšířenější metodou hledání podobností je pravděpodobně pomocí převodu dokumentů do modelu vektorového prostoru [8]. Dokument - text je poté reprezentován jako vektor $d = (w_1, w_2, w_n)$, kde

- d je vektor dokumentu
- w je termín dokumentu (v našem případě klíčová slova nebo atributy)
- n je počet termínů v dokumentu

Každé dimenzi odpovídá samostatný termín a pokud se v dokumentu vyskytuje je hodnota jeho vektoru nenulová. V oblasti informatiky a vyhledávání informací nabývá podobnost hodnotu z $[0; 1]$, kde 0 znamená, že dva vzorky si nejsou vůbec podobné a 1 značí identitu.

Výhody oproti Booleanovkém modelu (předchozí kapitola 4.2.1 jsou:

1. Možnost hodnocení dokumentů podle jejich relevance.
2. Částečná shoda dokumentů.
3. Přidělení vah atributům.
4. Možnost výpočtu podobnosti mezi dokumenty.

Nevýhodou pak může být, že příliš dlouhé dokumenty mají velmi malou podobnost z důvodu rozsáhlé dimenze prostoru, porovnávaná slova se musejí přesně shodovat s ostatními slovy dokumentů a kontextově podobné dokumenty nejsou rozpoznány jako podobné.

Porovnávacích metrik dokumentů je relativně mnoho, mezi nejznámější patří:

Cosine similarity [9]: Počítající úhel mezi dvěma vektory a definována vzorcem:

$$sim_{cos} = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}} \tag{3}$$

Jaccard Index [10]: Porovnávající podobnost dvou množin, definován vzorcem:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{4}$$

Dice coefficient [11]: Podobně jako předchozí porovnává dvě množiny atributů, definován vzorcem:

$$d(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{5}$$

kde A je množina termínu dokumentu A a B je množina termínů dokumentu B. Pro množinu $A = (a, b, c, d)$ a množinu $B = (c, d, e, f, g)$ dosazením např. do vzorce Jaccard indexu dostaneme výsledek:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|(c,d)|}{|(a,b,c,d,e,f,g)|} = \frac{2}{7} = 0,285 \tag{6}$$

Podobnost mezi množinami je tedy 0,285, což v procentech můžeme vyjádřit jako 28,5 % shoda.

Tf-idf: Tf-idf je zkratkou pro *term frequency-inverse document frequency* popsán Saltonem, Wongem a Yangem [8]. Jedná se o statistickou metodu, která reflektuje, jak moc důležité je slovo v rámci dokumentu, kolekce dokumentů nebo korpusu a často se využívá při získávání informací. Hodnota tf-idf proporčně roste s počtem výskytů slova v dokumentu, ale je kompenzována frekvencí slova korpusu. To pomáhá kontrolovat skutečnost, že některá slova jsou častější než jiná, např. stop slova.

Variace tohoto přístupu se využívá v mnoha vyhledávačích jako hlavní nástroj k hodnocení relevance dokumentů pro zadaný dotaz. Váhový vektor pro dokument je definován jako $v_d = [w_{1,d}, w_{2,d}, w_{3,d}, ..., w_{N,d}]^T$ kde

$$w_{t,d} = tf_{t,d} \cdot log \frac{|D|}{|\{d' \in D | t \in d'\}|} \tag{7}$$

a $t_{f,d}$ je četnost termínu t v dokumentu d (lokální parametr), $log \frac{|D|}{|\{d' \in D | t \in d'\}|}$ je inverzní četnost dokumentu (globální parametr), $|D|$ je celkový počet dokumentů v porovnávané množíně dokumentů a $|\{d' \in D | t \in d'\}|$ počet dokumentů obsahující termín t.

4.2.3 Jazykové modelování

Metoda jazykového modelování byla využita v článku a experimentu *Finding similar research papers using language models* [5]. Základní myšlenkou je odhadnutí unigramu jazkových modelů pro každý dokument (v tomto případě abstraktu) a výpočet jejich odlišnosti. Dokument *d* je pak považován za generovaný modelem D. Unigram je n-gram velikosti 1, což při aplikaci na text abstraktu v podstatě znamená rozdělit jej na jednotlivá slova. Pro ty se poté vypočítá pravděpodobnost v dokumentu, na kterou bývá ještě použita metoda vyhlazování (např. Laplacova, Jelinek-Mercerova), znázorněny jsou v tabulce 1. Pravděpodobnost, že model dokumentu vygeneruje slovo w, je dána vzorcem:

$$P^*(w|D) = \lambda P(w|d) + (1 - \lambda) P(w|C) \tag{8}$$

kde

Slova	Pravděpodobnost výskytu
je	0,1
rád	0,2
vidí	0,05
klika	0,03

Tabulka 1: Slova s pravděpodobností výskytu v dokumentu

- C je kolekce dokumentů
- λ je váha slova w

Problémem této metody je rovněž vyloučení synonym a souvisejících pojmů, který se nejvíce projevuje u krátkých textů, jakými jsou právě abstrakty. To může být částečně vykompenzováno přidáním porovnávání témat dokumentů. Protože ale nejsou známa, musí být odhadnuta například podle Latentní Dirichletovy Alokace (anglicky *Latent Dirichlet Allocation*[12], zkráceně LDA).

Základní princip LDA vychází z latentní sémantické analýzy, což je statistický přístup zpracovávání přirozeného jazyka sloužící k nacházení vztahů mezí dokumenty, jejich slovy, synonym, atd. LDA spouje tento přístup s Dirichletovou pravděpodobností a vychází z předpokladu, že každý dokument je složen z více témat a každé jeho slovo lze přiřadit alespoň jednomu z nich.

4.2.4 Analýza shlukováním

Shluková analýza [13] (angl. *cluster analysis*) je statistická metoda používající se ke klasifikaci objektů. Výstupem této analýzy jsou setříděné skupiny podobných dokumentů na základě jejich atributů (slov).

Metody shlukování se dělí na hierarchické a nehierarchické:

1. *Hierarchické* metody jsou jakýmsi větvením vedoucím k zjemňování klasifikace dokumentů. Jedná se o systém podmnožin, kde jejich průnikem je buďto množina prázdná nebo jedna z nich.
2. *Nehierarchické* metody vytvářejí shluky oddělené, průniky těchto množin jsou prázdné.

Existuje řada způsobů, jak shlukovat objekty (dokumenty), základními jsou:

- *metoda nejbližšího souseda*: Vzdálenost je určována dvěma nejbližšími objekty daných shluků
- *metoda nejvzdálenějšího souseda*: Vzdálenost je naopak určena dvěma nejvzdálenějšími objekty daných shluků
- *párová vzdálenost*: Vzdálenost se určuje průměrem vzdálenosti všech párů objektů mezi různými shluky, průměr může být vážený i nevážený

- *centroidní medota*: Vzdálenost je určena mezi odhadnutými středy daných shluků, může být vážená i nevážená.
- *Wardova metoda*: Slučuje shluky s minimálním součtem jejich čtverců. Vychází z analýzy rozptylu.

4.3 Částečná shoda textových řetězců

I přes správný výběr vhodných atributů k porovnání dvou profilů brzy narazíme na problém, kdy jedna a tatáž entita je popsána trochu jiným výrazem. Příkladem může být uvedení katedry informatiky VŠB na fakultě elektrotechniky. V rámci digitálních knihoven se vyskytují různé názvy pro stejnou katedru, např. výrazy *VŠB – Technical University of Ostrava* a *VSB Tech. Univ. of Ostrava*. Pouhým porovnáním řetězců, byť po odstranění nežádoucích znaků, bychom přišli o fakt, že jde o stejnou entitu, jen jinak zapsanou.

Budeme tedy muset použít algoritmus pro tzv. částečnou shodu (anglicky *fuzzy match*). Princip algoritmů je založen na měření počtu primitivních operací [14] potřebných k převedení jednoho textového řetězce na druhý, aby bylo dosaženo přesně shody mezi nimi. Tomuto počtu se říká *vzdálenost úprav* (angl. *edit distance*). Nejobvyklejší primitivní operace jsou:

- *vložení*: operace vložení znaku na jakoukoliv pozici textového řetězce (kop → ko*p*r)
- *substituce*: operace nahrazení jakéhokoliv znaku v textovém řetězci za jiný (kop*r* → kop*a*)
- *odstranění*: operace odstranění jakéhokoliv znaku v textovém řetězci (kop*r* → kop)
- *transpozice*: operace záměny znaku v textovém řetězci (*ko*p*o*r → *po*k*o*r)

Kromě hledání podobností textů se tyto principy používají také ke kontrole pravopisu, hledání podobnosti ve vzorcích DNA, filtraci spamu [15], identifikaci hudby podle krátkých útržků či korekce pro optické rozpoznávání znaků. Známé metriky pro výpočet vzdálenosti mezi řetězci jsou například:

Levenstheinova vzdálenost: Velmi známa a používaná metrika vymyšlena Vladimírem Levenstheinem [16] se stala také objektem mnoha modifikací. Původní výpočet je dán vzorcem:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{jestli } min(i,j) = 0 \\ \min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{jinak} \end{cases} \tag{9}$$

kde

- a a b jsou textové řetězce k porovnání

- $1_{(a_i \neq b_j)}$ je funkce vracející hodnotu 0, když $a_i = b_j$, jinak hodnotu 1

Levenstheinova vzdálenost nebere v potaz primitivní operaci transpozice. Pro dva textové řetězce *kopr* a *klopa* bude výsledkem hodnota 2 (jedno vložení a jedna substituce).

Damerau–Levenshteinova vzdálenost: Damerau–Levenshteinova vzdálenost [17] je modifikací předchozí Levenstheinovy metriky a navíc počítá také s primitivní operací transpozice, tedy záměny znaků v textech. Hraje důležitou roli zejména ve zpracování přirozeného jazyka a při porovnávání vzorků DNA.

Jaro-Winklerova vzdálenost[18]: Tato metrika je variantou původní Jaro vzdálenosti [19] a výsledkem je podobnost obou textových řetězců, kde *0* znamená žádnou podobnost a *1* naprostou shodu. Výpočet je dán následujícími postupy. Nejdříve se musí spočítat původní Jarova vzdálenost dle vzorce:

$$d_j = \begin{cases} 0 & \text{jestli } m = 0 \\ \frac{1}{3}(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}) & \text{jinak} \end{cases} \tag{10}$$

kde:

- *m* je počet shodných znaků v řetězcích (např. pro řetězce *Adolf* a *Adam* bude tato hodnota rovna *2*)
- *t* je polovina počtu transpozic v řetězcích (např. řetězce *ruka* a *kura* mají výměnu písmen r/k a k/r tedy $\frac{2}{2} = 1$)

Jaro-Winklerova vzdálenost je nadefinována jako:

$$d_w = d_j + (lp(1 - d_j)) \tag{11}$$

kde:

- d_j je Jarova vzdálenost (viz vzorec 10)
- l je délka společného prefixu porovnávaných řetězců, maximální hodnota může být *4*
- p je konstanta vyjadřující důležitost společného prefixu, maximální hodnota je *0,25*, standardně se používá *0,1*

Z výše uvedeného je patrné, že díky možnosti přiřazení váhy prefixu řetězců se dá tato metrika používat například pro kontrolu chyb v textu a překlepů.

4.4 Základní tvary klíčových slov

Schopnost správného rozpoznání stejných entit nestejně popsaných může být vylepšena o převedení jednotlivých slov do nějakého normalizovaného tvaru před porovnáváním částečným odhadem (popsáno v předchozí podkapitole). Slova jako *testování*, *tester*, *testy* by měly být převedeny na identický tvar *test*. Toho se dá dosáhnout na první pohled jednoduchým nalezením základního tvaru slov nebo jejich kořenu.

Základní tvar

Nalezením základního tvaru slov, tzv. lemmatizací dostaneme slovníkový tvar. Je velmi obtížné navrhnout algoritmus, který by toto strojově dokázal. Vychází se proto s předpřipravených slovníků - korpusů, která pro většinu slov definují jejich možné tvary a reversním postupem se dopracují k základnímu tvaru. Pro slova *běh*, *běhání*, *běhat* bychom měli správným postupem získat slovo *běh*.

Kořen slova

Hledáním kořene (anglicky *stemming*) slova nemusíme získat správné pravopisné slovo, ale jen společnou základní část slova. Návrh automatizovaných algoritmů je v tomto případě jednodušší a pro anglický jazyk jich existuje hned několik.

První úspěšnější a velmi rozšířený algoritmus navrhl v roce 1980 Martin Porter [20], o několik let později jej zdokonalil a vznikl prozatím nejpřesnější stemmer pod označením *Snowball*. Tento algoritmus se také podařilo rozšířit o podporu českého jazyka, postup je popsán v článku *Nalezení slovních kořenů v češtině* [21].

Nevýhodou nalezeného kořene slova je, že může být identický pro slova zcela jiného významu. Odstraněním předpon a přípon navíc může dojít ke zkreslení a ztrátě původní informace o slovu.

5 Hledání identit uživatelů na sociálních sítích a digitálních knihovnách

Kapitola se věnuje návrhu algoritmu pro hledání identit uživatelů na sociálních sítích a digitálních knihovnách. U některých podkapitol jsou také uvedeny poznámky týkající se implementace experimentálních algoritmů.

5.1 Komunikace s vnějším světem

Pro správnou funkčnost algoritmu je zapotřebí vytvořit spolehlivé nástroje stahující data z externích stránek a služeb, v našem případě ze sociálních sítí a digitálních knihoven. Bohužel ke komunikaci s prvním jmenovaným bude muset přistupováno individuálně, každá sociální síť nabízí jiné možností přístupu a hlavně poskytuje různá data vhodná k porovnání. To bude také zohledněno při dalším návrhu algoritmu. Důležité je, aby pro každá síť umožňovala hledání uživatele podle jeho jména a příjmení. Všechny služby, které toto neumožňují, jsou pro algoritmus nevhodné.

5.1.1 Komunikace s digitálními knihovnami

Digitální knihovny poskytují data v podobném formátu, přístup k nim se liší jen v metodě jejich získání. Servery SpringerLink a IEEExplore umožňují přistupovat k meta-datům pomoci veřejného API. U prvního jmenovaného je bohužel přístup omezen na určitý počet požadavků za vteřinu, dotazy bude tedy vhodné ukládat do cache a v případě opakovaného dotazu získat data z ní. Knihovna ACM žádné API neposkytuje, data tedy budou stahována přímo z webu pomocí crawleru (automatického stahovače) a získána přímo ze zdrojového HTML kódu a výsledky se budou opět cachovat.

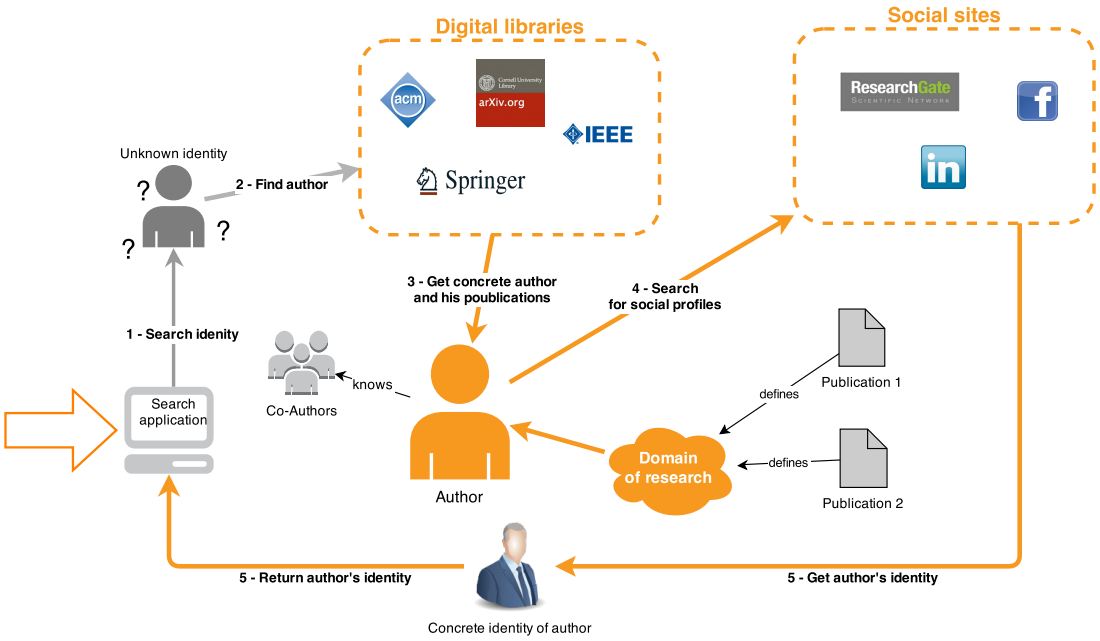
5.1.2 Komunikace se sítí LinkedIn

Síť LinkedIn poskytuje veřejné API pro komunikaci pomocí REST a OAuth protokolů, bohužel neumožňuje vyhledávání z bezpečnostních důvodů chránící uživatele služby (pozn. tento důvod byl uveden na webu LinkedIn). K účelům výzkumu této diplomové práce byla udělena výjimka a komunikace aplikace bude probíhat přes zmíněné API.

Jako alternativa tohoto přístupu může být čtení informací přímo ze zdrojového kódu stránek, LinkedIn nabízí veřejně dostupné vyhledávání se základními informacemi uživatelů. Počet výsledku je ale omezen na 20.

5.1.3 Komunikace se sítí Researchgate

Služba Researchgate neposkytuje veřejné API, které by usnadňovalo komunikace služeb třetích stran. Situaci navíc komplikuje fakt, že web je funkční pouze se zapnutým Javascriptem, což znemožňuje získávání informací pomocí klasického crawleru posílající HTTP požadavky. Bude proto využitu projektu Selenium (viz podkapitola 5.6) pro jazyk Python. Selenium je zjednodušeně řečeno zautomatizovaný prohlížeč, kterým je možno



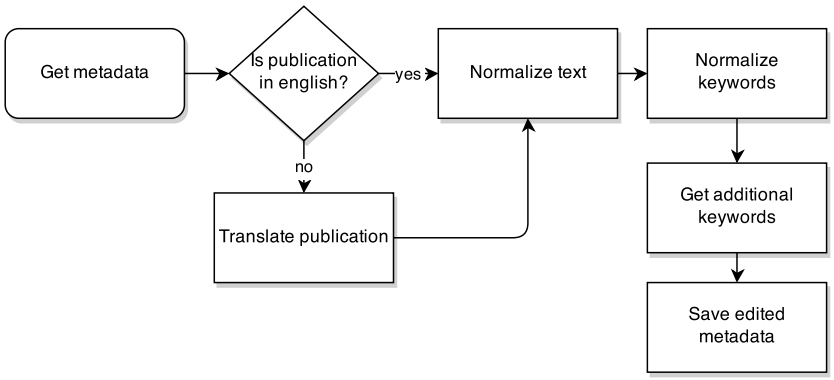
Obrázek 5: Komunikace s vnějším světem

ovládat a simulovat chování pomocí příkazů a získávat z něj data ve strojově čitelné podobě i s podporou Javascriptu a jiných interaktivních technologiích.

5.2 Zpracování publikací

Protože základní myšlenkou této práce je hledat identity a dále pracovat s uživateli primárně na základě jeho publikační činnosti, je potřeba tyto publikace získat a vhodně upravit pro další práci s nimi. V prvním kroku je tedy potřeba získat a následně analyzovat meta-data z digitálních knihoven. V aplikaci je potřeba mít implementované nástroje pro komunikace s digitálními knihovnami popsanými výše, které vrací tyto informace o publikaci:

- *Titulek* - titulek/název publikace ve formátu nalezeném v knihovně
- *Autoři* - seznam autorů rozdělených na příjmení a jméno, u jména se počítá také s variantou počátečního písmena
- *Affiliates* (instituce) - seznam institucí, na kterých působí autoři; slouží k lepšímu odhadu identity uživatele
- *Klíčova slova* - tzv. keywords uvedena u publikace, ne vždy ale bývají zadána
- *Rok vydání* - rok, kdy byla publikace vydána; slouží pro porovnání publikací v závislosti na čase



Obrázek 6: Návrh algoritmu zpracování publikace

- *Abstrakt* - abstrakt uvedený u publikace; bývá zadán všude, bohužel ne vždy je dostatečně dlouhý a vhodně napsaný; používá se pro získání dalších klíčových slov

Problémem je, že tato data nejsou sjednocena napříč digitálními knihovnami, dokonce nebývají jednotná ani v rámci jedné služby. Autoři jsou uváděni někdy s celým jménem a příjmením, jindy jen s počátečním písmenem jména. U českých jmen také některé knihovny nepracují s diakritikou. Instituce jsou psané také různě v závislosti, jak je vyplnili autoři. U klíčových slov se dále musí počítat také s hledáním synonym a normalizovat jejich tvar.

Překlad publikace

Z důvodů analýzy textu, zpracování abstraktu a použitých nástrojů je nutno přeložit neanglické publikace do angličtiny. Pro tyto účely byl vybrán online překladač Google Translate, který dokáže rovněž velmi spolehlivě detekovat jazyk textu. Překlad probíhá pomocí REST API obsaženého v knihovně NLTK, zdarma je k dispozici překlad pro jeden milion znaků, další jsou již zpoplatněny.

Pozn. autora: V následujícím textu může být použit anglický text z důvodu zachování principů.

Normalizace textu

Všechna získaná data se převedou na malá písmena a odstraní se z textu závorky, nadbytečné mezery a nežádoucí znaky (pomlčky, podtržítka, ...). Pro jistotu se navíc zkontroluje text, jestli neobsahuje špatně zformátovanou diakritiku a převede se eventuálně na text bez ní. Nyní je text vhodný k porovnání odhadem popsaným v kapitole 4.

Normalizace klíčových slov

Klíčová slova projdou stejně jako veškerý text předchozím procesem. Problémem ovšem stále zůstává porovnávací schopnost se slovy jiných publikací. Mějme příklad s následujícími klíčovými slovy: *testing software*, *testing application*, *checking software*. Je evidentní,

že významově se jedná o stejné slovo, pouhým porovnáním textových řetězců bychom ovšem tuto skutečnost nepotvrdili. Pro každé klíčové slovo tedy vytvoříme seznam možných kombinací synonym s využitím slovníku WordNet a knihovny NLTK pro Python.

Pro každé slovo nalezneme jeho základní tvar takzvanou lemmatizací (viz kapitola 4). Složená klíčová slova setřídíme podle abecedy pro rychlejší následující porovnání.

Získání dodatečných klíčových slov

Protože ke každé publikaci není uveden dostatečný počet klíčových slov a ne vždy jsou vhodně zvolená, je nutno analyzovat abstrakt a pokusit se z něj vytáhnout vhodná slova. Celý postup je znázorněn v následujícím pseudokódu:

```
dodatacnaKlicovaSlova = array()
tokeny = Tokenizuj(abstrakt)
tagy = NajdiSlovniDruhy(tokeny)
pojmenovaneTokeny = RozpoznejPojmenovaneEntity(tagy)

for token in pojmenovaneTokeny:
    if token != osoba nebo geo:
        odstranTokenZanalyzovanych(token)

for tag in tags:
    if tag je negator:
        oznacTagJakoNeg(tag)

vsechnyFraze = najdiFrazePodstanychJmen(tags)

for fraze in vsechnyFraze:
    if fraze je pozitivni fraze:
        fraze = Lemmatizuj(fraze)
        fraze = OstranStopSlova(fraze)
        if KlasifikujFrazi (fraze) == True:
            dodatecnaKlicovaSlova.add(fraze)
return dodatecnaKlicovaSlova
```

Výpis 1: Pseudokód nalezení dodatečných klíčových slov

Nejdříve vytvoříme tzv. *tokeny*, což znamená, že rozdělíme abstrakt na jednotlivá slova a nalezneme pro ně slovní druhy pomocí NLTK knihovny. Tento seznam bude poté použit jako vstup do funkce rozpoznávající pojmenované entity. Za pomoci předpřipraveného korpusu odhadne knihovna pro jednotlivé slovní druhy dle klasifikátoru, zda se jedná o *osobu*, *čas*, *místo*, *geo-politika*, *lokalitu*. Nás zajímají hlavně klíčová slova označena jako *osoba* a *geo-politika*, protože během analýzy desítek zpracovaných abstraktů bylo zjištěno, že pro nás zajímavá klíčová slova bývají označována právě těmito jmény. Jinak pojmenovaná slovní spojení by mohla zkreslovat výsledky, proto je pro další zpracování zahodíme.

Zbývající nalezená slova z abstraktu znovu zkontrolujeme a pokusíme se nalézt ta, která způsobují negaci ve větě. Potřebujeme eliminovat všechna spojení, která jsou těmito spojkami a slovesy významově vyloučena. Mějme příklad věty *Zabývá se návrhem databází, ale ne optimalizací dotazů.* Z této věty bychom mohli získat klíčová slova *návrh databáze*

a *optimalizace dotazů*, je ale jistě zřejmé, že o druhý případ v textu vlastně vůbec nejde, protože byl vyloučen spojkou *ale*. Označíme ji tedy přívlastkem "NEG"pro následující vyloučení.

Následuje krok pro nalezení frází podstatných jmen. Během analýzy mnoha abstraktů bylo opět zjištěno, že takto označená spojení slov jsou dostačující a relativně vhodná k výběru za klíčová. Jejich nalezení je snadné pomocí následujících regulárních výrazů:

$$\text{NP} : \{< JJ|NN.* > \}$$

pro pozitivní fráze podstatných jmen a

$$\text{NP-NEG} : \{< *NEG >< DT|JJ|NN.* > +\}$$

pro negativní fráze podstatných jmen.

Zjednodušeně řečeno, nalezneme všechny skupiny sousedních slov přídavných a podstatných jmen a sloučíme je do jediného slova. Ve variantě negativních frází nalezneme negovací slovo, označíme vše za nim, a tyto negované skupiny vyloučíme z další analýzy. Jde o poměrně jednoduché řešení, ale zároveň také efektivní. Můžeme tak sice přijít o potencionální vhodná klíčová slova, ale toto riziko je přístupné z důvodu dalších mnoha vhodných slov v textu. Navíc výskyt negací není v abstraktu až tak běžný a, jak se ukázalo, je lepší zahodit raději možný vhodný termín než vybrat ten, co měl být vyloučen. Nedochází poté k takovému zkreslení informací a špatnému přiřazení publikací autorům.

Každému slovu z pozitivní fráze poté nalezneme základní tvar slova a opakujeme proces normalizace klíčových slov. Dále odstraníme stop slova z předem připraveného seznamu a ověříme si jeho vhodnost klasifikátorem.

Pro tyto účely byla zvolena implementace Naivního Bayesova klasifikátoru [22] obsažena v NLTK knihovně. Jedná se o jednoduchý pravděpodobnostní klasifikátor založený na Bayesově teorému [23]. Pro správné vyhodnocování je nutné mít připravenou trénovací množinu, na základě kterých poté určuje pravděpodobnost vstupních dat a rozhoduje o jejich správnosti. Byla vytvořena množina čítající cca 5500 klíčových slov a manuálně zkontrolována a označena vhodná a nevhodná slova.

Pokud je tedy klíčové slovo vyhodnoceno jako vhodné, je přidáno do seznamu dodatečných slov.

5.3 Hledání autora v digitálních knihovnách

Zpracováním abstraktu jsme získali data vhodnější pro hledání a porovnávání profilů uživatelů, v našem případě autorů. Princip algoritmu po identifikaci osoby založíme na porovnání a hledání podobnosti mezi *obecnými* atributy. Ideální by porovnat alespoň jednu *unikátní* vlastnost, bohužel ale zvolený přístup vycházející z publikací žádnou takovouto možnost neumožňuje.

Podrobnějším zkoumáním publikací a možností spárování vychází jako nejlepší možnost identifikovat výzkumníka podle jeho spoluautorů a institucí. Porovnáváním pouze na základě klíčových slov, respektive jakékoliv jiné reprezentaci obsahu samotné publikace, bychom nemohli správně sjednotit identitu autora v rámci více publikací. Mnoho

výzkumníků publikuje v mnoha různorodých oblastech výzkumu, typickým příkladem může být situace doktorandů a jejich vedoucích, kteří spolu publikují články ve více oblastech vědy.

Nástin samotného algoritmu je znázorněn v pseudokódu 2.

```
jmeno, prijmeni, [spoluautori] <- {vstup uzivatele}
nalezenePublikace = array()
publikace = array()
for hledac in hledaciKnhioven:
    nalezenePublikace += hledac.Hledej(jmeno, prijmeni)

sjednocenePublikace = SjednotStejnePublikace(nalezenePublikace)
seskupenePublikace = SeskupPodleAutoruInstituci(sjednocenePublikace)

for publikace in seskupenePublikace:
    publikace += AnalyzujPublikace(publikace)

for pub1 in Length(publikace) == 1:
    for pub2 in publikace
        podobnost = Porovnej(pub1, pub2)
        if podobnost > prahovaHodnota:
            Seskup(pub1, pub2)

PridejDoDatabaze(publikace)
Opakuj pro spoluatory
```

Výpis 2: Pseudokód nalezení identity uživatele na digitálních knihovnách

Nejdříve požádáme uživatele systému o zadání jména a příjmení autora k hledání. Tyto dva údaje jsou nezbytným a minimálním předpokladem pro správnou funkčnost algoritmu a slouží k dalšímu rozhodnutí o identitě. Tyto údaje jsou poté předány jednotlivým vyhledávačům, tedy digitálním knihovnám IEEE, ACM a SpringerLink. Vyhledávače vrátí výsledky v podobě vytvořených instancí totožné třídy, vložíme je tedy do stejného pole *nalezenePublikace*.

V následujícím kroku musíme sloučit stejné publikace nalezené v různých digitálních knihovnách, což je poměrně častý jev. Publikace mezi sebou vzájemně porovnáme pomocí algoritmu částečné shody (viz kapitola 4) podle názvu publikace, roku vydání a spoluautorů. Důvodem zahrnutí roků vydání a spoluautorů je výskyt více publikací pod stejným jménem. Ač se tak neděje velmi často, navíc u stejného autora, pro jistotu je toto ověření zahrnuto. Během sloučení je potřeba také porovnat nalezená metadata publikací, ne všechny knihovny vracejí všechna metadata. Doplníme je tedy napříč jednotlivými výsledky stejné publikace, ať máme k dispozici co nejobsáhlejší soubor metadat o publikaci.

Zatím máme stále jen soubor publikací zmenšený o duplicity, pokusíme se tedy o první odhad autorů. V dalším kroku vytvoříme skupiny publikací dle spoluautorů a institucí publikací. Každou publikaci porovnáme s již nalezenými skupinami a to následujícím způsobem:

- Ze všech autorů publikace vyjmeme hledaného autora.

- Takto získané spoluautory porovnáme se spoluautory již vytvořených skupin.
- Stejným způsobem porovnáme instituce publikací a již vytvořených skupin.
- Ke přidání publikace do porovnávané skupiny dojde, pokud:
  - Se shoduje alespoň jedna instituce, nebo
  - se shoduje alespoň jeden spoluautor a jedna instituce, nebo
  - se shodují alespoň dva spoluautoři
- V opačném případe se z publikace vytvoří nová skupina.

Dostáváme první možné identity uživatelů v rámci digitálních knihoven. Předchozí postup funguje dobře v případě, že nalezené publikace neobsahují žádnou, která by měla pouze jediného autora. Pro tento případ se pokusíme odhadnout totožnost autorů porovnáním podobnosti publikací. Ze seskupených publikací vybereme všechny, které obsahují pouze jednoho autora, a každou z nich porovnáme se skupinami ostatními, respektive s jejich publikacemi. Využívá se metodiky porovnávání v modelech vektorového prostoru (kapitola 4), konkrétně pomocí Jacardova indexu (viz vzorec 4). Vhodné je volit prahovou hodnotu vyšší, velké množství publikací v porovnávané skupině může vést ke zdánlivé podobnosti pomocí několika mála nalezených klíčových slov, která se ale mohou vyskytovat ve více oblastech výzkumu a vědy, a nemusí tedy zapadat přímo do oblasti výzkumu daného uživatele. Důležitou roli zde hraje také atribut roku vydání. Pokud by mezi dvěma publikacemi byl časový rozdíl roku publikace více než 65 let, pravděpodobně se již nebude jednat o jednoho autora. V úvahu připadá samozřejmě také mnohem nižší rozdíl, ovšem např. doba 50 let by neměla definitivně rozhodnout, ale jen znevýhodnit výsledek k rozhodnutí. Autor může publikovat i v 75 letech (a samozřejmě se tak i děje).

Nyní máme vytvořeny možné identity uživatelů. Zbývá je porovnat s údaji již uloženými v databázi, upravit existující o nově nalezené údaje, přidat nové publikace a spoluautory. Celý tento postup budeme opakovat ještě pro nalezené spoluautory. Zde je již odhad uživatelů značně usnadněn, vstupem není jen jméno a příjmení hledané osoby, ale také seznam osob vyskytujících se v jeho sociální síti. Budeme tedy ihned eliminovat nalezené profily, které tato spojení neobsahují.

Protože vstupními daty byly pouze jméno a příjmení, získali jsme odhadnuté profily autorů na základě jejich publikací (výsledek z implementované aplikace je znázorněn na obrázku 7). Tento vstupní krok je vhodné obohatit ještě o možnost určení klíčových slov autora, například by to mohla být slova *information retrieval* nebo *data mining* pro určení působnosti v oblastí získávání dat v informatice. Efektivní je také dodat seznam kolegů (spoluautorů) nebo instituci, na které uživatel působí, potažmo publikuje.

Poznámky k implementaci: Uživatel systému vyhledá autora pomocí jednoduchého formuláře. K zadanému jménu a příjmení vybere digitální knihovny, na kterých se má autor vyhledávat. V případě digitální knihovny ACM vzniká dlouhá časová prodleva ke zpracování výsledků, která je zapříčiněna nutností vytváření pauz mezi dotazy. ACM

Obrázek 7: Nalezené skupiny autorů pro vstup *John Doe*. Skupiny jsou vytvořeny podle spoluautorů a porovnáním klíčových slov publikací.

při častém a rychlém dotazování blokuje dočasně zdrojovou IP adresu a je tedy nutné částečně simulovat chování běžného uživatele.

Každá knihovna má vytvořenou svou vlastní třídu rozšiřující třídu *BaseSearcher* (naznačeno na obrázku 11). Funkce *search* této třídy po zavolání provede vyhledávání v rámci implementované knihovny a vrátí výsledky jako seznam instancí třídy *SearchItem*. Tímto návrhem je zajištěna snadná rozšířitelnost o vyhledávání na dalších digitálních knihovnách.

5.4 Hledání identit na sociálních sítích

Různé specializace a zaměření sociálních sítí jsou jedním z hlavních důvodů jejich rozšířenosti. Uživatelé internetu si mohou najít a používat pouze vybrané služby řešící funkce a problémy, které jsou jim sympatické, a nejsou omezeni pouze na užívání sociálních sítí vzniklých se záměrem komunikace s jinými uživateli (Facebook, Twitter). Bohužel tento fakt značně znemožňuje vytvoření jednotného způsobu hledání identit uživatelů dle univerzálního řešení. Každá sociální síť sbírá různá data o svých uživatelích a rovněž jejich aktivita není stejnoměrně rozložena napříč všemi službami. Dochází tedy k nehomogenní a neaktuální množině atributů, především těch časově závislých (ve vztahu, ženatý, případně bydliště) na profilech jednoho uživatele.

Z výše popsaných důvodů je zapotřebí přistupovat ke každé sociální síti individuálně a zmapování informacích na veřejných profilech je tedy nezbytné. Na základě této rešerše zajisté narazíme na sociální služby, které nejsou strojově zpracovatelné z nedostatku dostupných a vypovídajících informací, nebo naopak nalezneme profily lehce přiřaditelné k hledané osobě. V následujícím textu je návrh algoritmů pro dvě vybrané sociální sítě LinkedIn a ResearchGate.

5.4.1 LinkedIn

Hlavní myšlenkou hledání uživatelů na profesní sítí LinkedIn je porovnávání obecných atributů. Veřejný profil zobrazitelný bez nutnosti přihlášení, obsahuje informace, které mohou být stejné slučitelné s informacemi získanými z metadat publikací, a sice:

- *Jméno* a *příjmení*
- *Zaměstnání* - uživatel zpravidla uvádí všechny (nebo většinu) pozic, na kterých pracoval. Tyto pozice obsahují mimo popis samotné pozice také dobu výkonu, název firmy a lokalitu. V tomto seznamu se mohou vyskytovat instituce publikací.
- *Spojení* - uživatel se spojuje s ostatními uživateli zejména za účelem rozšíření své sítě kontaktu (angl. *networking*), čímž zvyšuje šance na oslovení a získání lepší pracovní pobídky. V těchto spojeních se mohou vyskytovat spoluautoři publikací.
- *Vzdělání* - uživatelova dosažená vzdělání, opět obsahují informace o době a lokalitě studia; mohou se zde vyskytovat autorovy instituce.
- *Publikace* - vědečtí výzkumníci často na svých profilech zveřejňují jimi vydané publikace; ač nejde o kompletní výčet, jedná se o velmi cenný atribut určující identitu uživatele téměř 100 %.
- *Dovednosti* - uživatel na svůj profil vyplňuje své schopnosti a dovednosti. Analýzou těchto seznamů můžeme najít určitou podobnost mezi klíčovými slovy jeho publikací.

Hrubý návrh funkce pro hledání uživatele na LinkedInu je znázorněn v pseudokódu 3.

```
vybranyUzivatel <- {vybrany uzivatel}
nalezeniUzivatele = NajdiLinkedinUzivatele(vybranyUzivatel)
mozniUzivatele = array()

for uzivatel in nalezeniUzivatele:
    NajdiAtributyNaProfilu(uzivatel)
    if ObsahujeZkusenosti(uzivatel):
        uzivatel.klicovaSlova = zkusenosti
    vysledek = PorovnejUzivatele(vybranyUzivatel)
    if vysledek > prahovaHodnota:
        mozniUzivatele += uzivatel

if mozniUzivatele:
    SetridUzivatele(mozniUzivatele, 'sestupne')
    return mozniUzivatele
else:
    return null
```

Výpis 3: Pseudokód nalezení identity uživatele na LinkedInu

Vstupním předpokladem algoritmu je uživatel s přiřazenými publikacemi uloženými v databázi. V síti LinkedIn vyhledáme všechny profily obsahující jeho jméno a příjmení a nalezené výsledky zpracujeme. Na webové stránce profilu nalezneme výše zmíněné atributy, a pokud o sobě uživatel uvedl zkušenosti, označíme je jako klíčová slova uživatele. Aktuálně zpracovávaný profil poté porovnáme s vybraným uživatelem. Proces porovnávání je opět založen na hledání podobností mezi modely ve vektorovém prostoru dle vah jednotlivých atributů podle vzorce:

$$sim_{u,p} = \begin{cases} \sum_{i=0}^{n} w_i \cdot sim(a_{i,u}, a_{i,p})) & \text{pokud } sim(a_{name}) > th_{name} \\ 0 & \text{jinak} \end{cases} \tag{12}$$

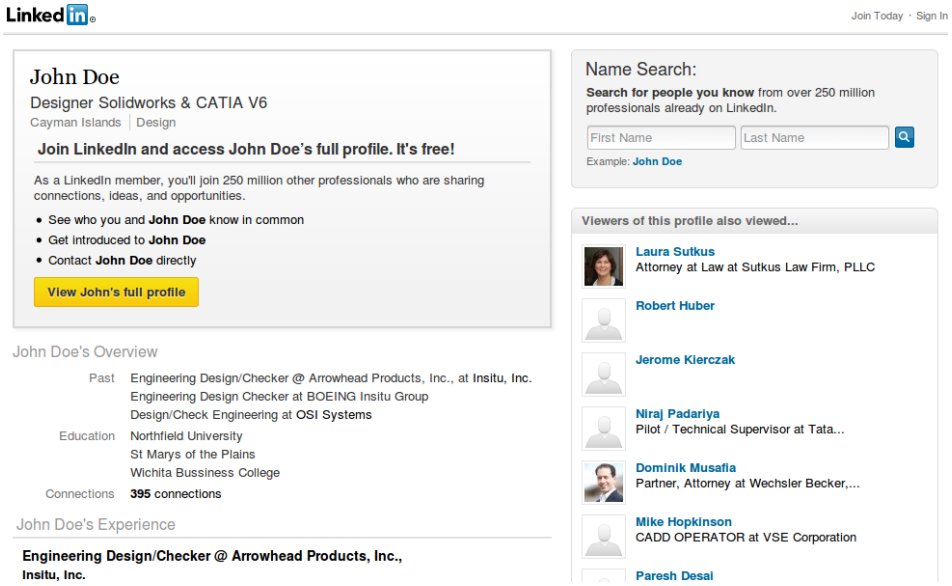
kde

- $sim(a_{name})$ je podobnost mezi jmény
- th_{name} je prahová hodnota k určení, zda se jména shodují
- n je počet porovnávaných atributů
- w_i je váha porovnávaných atributů
- a_p je množina atributů profilu uživatele na síti LinkedIn
- a_u je množina atributů hledaného uživatele v databázi
- $sim(a_{i,u}, a_{i,p})$ je podobnost atributů mezi profilem a vybraným uživatelem

Jak je z výše popsaného patrné, hlavním kritériem je samozřejmě ověření, že jméno hledaného uživatele se shoduje se jménem nalezeným na profilu. Doporučená hodnota porovnávající prahové hodnoty th_{name} je co nejbližší 1, např. $0,96$. Tím zajistíme případné překlepy ve jméně. Tento přístup se hodí zejména u dlouhých jmen a složitěji psaných jmen, kde snáze dojde k nechtěnému překlepu. Pokud je tedy podobnost jmen dostatečná, porovnáváme mezi sebou i další atributy. V opačném případě další podobnosti nehledáme a profil vyhodnotíme jako nevyhovující. Váha atributu publikací je zvolena jako nejvyšší a je vhodné ji zvolit okolo 0.9. Konkrétně v tomto případě se osvědčí uměle navýšit hodnotu podobnosti, pokud nějaká existuje. Můžeme přičíst konstantu (např. $0,1$), jestli je menší než 1. Tím dodáme publikacím ještě větší význam, což je také naším cílem. Je velmi pravděpodobné, že uživatelova publikace bude uvedena na jeho profilu a zároveň ji máme přiřazenou i v našem systému.

Váha pro spojení je zvolena také relativně vysoká, cca $0,65$, a pro instituce pak podobná hodnota $0,55$. Tyto hodnoty vypovídají o autorově síti kontaktů a ukázaly se býti dosti určujícími pro uživatelovu identitu.

Dovednosti, resp. klíčová slova, uživatele slouží především pro ověření a potvrzení nejistého odhadu. Přiřazení příliš velké váhy by způsobovalo zkreslení celkového skóre podobnosti, nelze výsledek určit pouze na základě těchto atributů, na druhou stranu podobnost nalezená mezi klíčovými slovy publikací a dovednostmi uživatele nebývá vysoká, váhovou hodnotu je dobré volit okolo $0,3$.



Obrázek 8: Profil uživatele sítě LinkedIn

Porovnáním profilů tedy získáme seznam potencionálních identit hledaného uživatele. Pokud obsahuje více nalezených výsledků, pro přehled jej ještě setřídíme sestupně podle skóre.

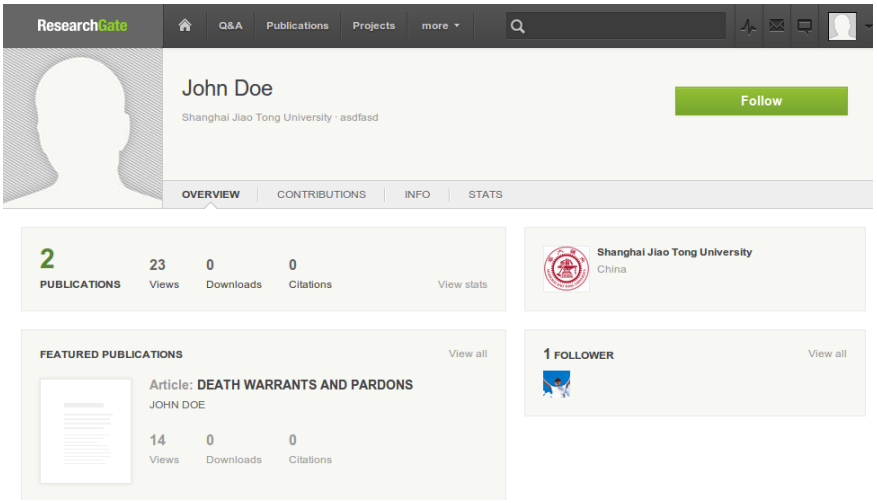
Pokud potvrdíme správnost nalezeného profilu, stejný postup zopakujeme pro všechny spoluautory uživatele. Situace je zjednodušena o fakt, že známe spojení na LinkedInu a průnikem jmen spoluautorů získáme konkrétní uživatelské profily bez nutnosti další manuální verifikace (samozřejmě pokud jsou profily podobné).

5.4.2 ResearchGate

Koncept služby ResearchGate zaměřující se hlavně na výzkumníky a struktura uživatelských profilů velmi připomíná digitální knihovny obohacené o rozměr aktivity uživatele. Zvolíme tedy přístup, který bude velmi podobný hledání identity na síti LinkedIn (viz předchozí podkapitola) a digitálních knihovnách. Porovnávání profilů s uživatelem je opět založeno na principu podobnosti atributů, a sice *unikátních* a *obecných*. Mezi unikátní atributy zařadíme publikace. Pokud ověříme jejich názvy, rok vydání a případně další dostupné informace, můžeme o nich říci, že se pro stejného autora již nebudou nikde vyskytovat.

Za obecné považujme opět spojení, resp. sledování, s jinými uživateli, kteří budou kandidáty na spoluautory publikací. Ověřovány budou také instituce a rovněž disciplíny nahrazující klíčová slova.

Princip se velmi podobá přístupu sociální sítě LinkedIn, postup je uveden ve výpisu 4.



Obrázek 9: Veřejný profil uživatele na síti ResearchGate

```
vybranyUzivatel <- {vybrany uzivatel}
nalezeniUzivatele = NajdiResearchGateUzivatele(vybranyUzivatel)
mozniUzivatele = array()

for uzivatel in nalezeniUzivatele:
    NajdiAtributyNaProfilu(uzivatel)

    if ObsahujeZkusenosti(uzivatel):
        uzivatel.klicovaSlova = zkusenosti
    vysledek = PorovnejUzivatele(vybranyUzivatel)

    if vysledek > prahovaHodnota:
        mozniUzivatele += uzivatel

if mozniUzivatele:
    SetridUzivatele(mozniUzivatele, 'sestupne')
    return mozniUzivatele
else:
    return null
```

Výpis 4: Pseudokód nalezení identity uživatele na ResearchGate

Opět se porovnávají atributy hledaného uživatele s atributy nalezených profilu podle vzorce 12. Jediný větší rozdíl je ve výpočtu podobnosti mezi publikacemi. V případě ResearchGate se získaná hodnota dále neupravuje a pouze se vynásobí váhou atributu, která i v zde zůstává vysoká ($w \approx 0,9$). Ostatní váhy je možné také zachovat jako u přístupu LinkedIn.

Protože je ResearchGate kombinací digitálních knihoven a uživatelé zde zveřejňují svou publikační činnost, nalezené publikace, které ještě nemáme přiřazeny k uživateli, dodatečně přidáme.

Stejně jako u sítě LinkedIn, i zde porovnáme profily spoluautorů s nalezenými spojeními uživatele na ResearchGate.

5.4.3 Facebook, Google+

Sociální sítě Facebook a Google+ nebyly zahrnuty do implementace a experimentu z důvodu neschopnosti zobrazit profily cizích uživatelů s dostatečným množstvím informací. Princip by ale mohl fungovat podobně jako v případě sítě LinkedIn. Uživatelé obou sítí se opět seskupují s jinými lidmi pomoci spojení (u Facebooku přátele v případě Google+ sociální kruhy), které by mohly reflektovat spoluautory publikací. Na obou sítích uživatelé nezřídka uvádějí informace o svých aktuálních zaměstnáních, což opět napodobuje vztah mezi institucemi autorů publikací.

Podobnost těchto dvou sítí je značná a obě požadují po uživateli vytváření stejné aktivity. Bohužel činnost uživatele se pak přesouvá pouze na jednu sociální síť, nebo ji alespoň upřednostňuje. Potlačená služba pak obsahuje neúplné nebo neaktuální informace, které nemusí být dostatečné pro správné nalezení profilu hledaného uživatele. Tohoto faktu se dá ovšem využít. Pokud nalezneme profil uživatele na jedné sociální síti, např. na Facebooku, získáním dalších unikátních a obecných atributů z profilu můžeme provézt porovnání s výsledky druhé sociální sítě, Google+. Situace je nastíněna na obrázku 4. Pouhým porovnáním unikátního atributu telefonního čísla, jména osoby a případným ověřením například pohlaví, získáme téměř jistě pravdivé spojení dvou profilů uživatele na různých sociálních sítích.

Poznámky k implementaci: Vyhledávání informací na sociální síti ResearchGate si vyžádalo speciální přístup. Sběr dat je řešen za pomocí projektu Selenium (viz podkapitola 5.6) z důvodu nefunkčnosti webových stránek bez povoleného Javascriptu. Interakce vyhledávače s služby ResearchGate tak probíhá přímo v prohlížeči. Rovněž je nutno mít nastaveny funkční přihlašovací údaje kvůli vyhledávání uživatelů sítě, samotné stahování informací z profilů již probíhá anonymně. Stejně jako v případě digitální knihovny ACM, i zde se simuluje chování běžného uživatele, což má za následek zpomalení zpracování výsledků.

V případě sítě LinkedIn je situace komplikována faktem, že v průběhu práce na tomto projektu služba výrazně omezila dostupné informace na veřejných profilech svých uživatelů. Stalo se tak tomu po dokončení experimentu a implementovaný vyhledávač pro LinkedIn byl dodatečně upraven na nové podmínky. Je ovšem možné, že získání dat z některých profilů nemusí fungovat správně.

5.5 Doporučování publikací

Autor zpravidla publikuje články v rámci své oblasti výzkumu, na základě tohoto předpokladu založíme myšlenku doporučování pro uživatele zajímavých publikací. Zpracováním publikace podle popsané výše v kapitole 5.2 získáme množinu klíčových slov popisující danou publikaci. Sloučením všech těchto množin autorových publikací pak zís-

káme doménu výzkumu (model), která bude vstupními daty pro porovnávání s publikací ve vektorovém prostoru, viz výpis 5.

```
vybranyUzivatel <− {vybrany uzivatel}
vybranePublikace <− {vybrane publikace}
domenaUzivatele = array()

for publikace in vybranyUyivatel.vsechny_publikace:
    domenaUzivatele += publikace.klicovaSlova

vhodnePublikace = array()

for publikace in vybranePublikace:
    skore = Podobnost(domenaUzivatele, publikace.klicovaSlova, [publikace.rokVydani])
    if skore > prahovaHodnota:
        vhodnePublikace += publikace
```

Výpis 5: Pseudokód doporučení publikací

Pro vybraného uživatele nejprve zjistíme doménu výzkumu seskupením všech klíčových slov jeho publikací. Duplicity slov napříč publikacemi zanecháme z důvodu zdůraznění, pokud se např. klíčové slovo *data mining* vyskytuje ve třech publikací z šesti, zřejmě se jedná o dlouhodobější zájem autora o tuto oblast.

Po vytvoření domény je nutno projít soubor doporučovaných publikací a vzájemně je porovnat. Děje se tomu podle již zmíněných metrik v kapitole 4.2, a sice kosínové podobnosti (vzorec 3), Jaccardova indexu (vzorec 4) a Diceova koeficientu (vzorec 5). Pokud je podobnost mezi doménou autora a publikací větší než zvolená prahová hodnota, zařadíme ji do seznamu možných vhodných publikací.

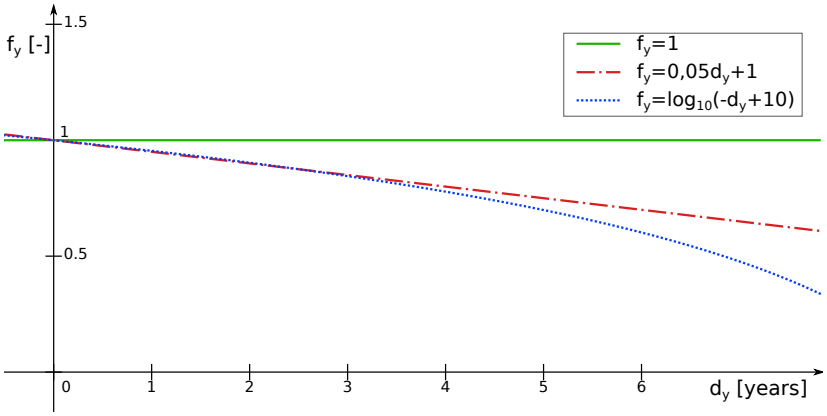
Jsou obory, kde nabyté vědomosti, provedené výzkumy a teorie velmi rychle ztrácejí na aktuálnosti, stárnou a zažité metody nahrazují nové a efektivnější. Typickém příkladem budiž počítačové vědy a informatika. Technologie pokračují rychlým tempem a potřeba neustálého vzdělávání je zde více než nutná. Tento fakt reálného světa zohledníme při procesu posuzování vhodnosti publikace k autorovi (případně naopak). Pokud výzkumník publikoval článek před deseti lety, nemusí být jeho zájem i stejnou problematiku již aktuální, na druhou stranu je potřeba vzít v potaz, že se kdysi o tuto oblast zajímal a přeci jen by ještě mohl mít v dané problematice přehled, byť menší.

Zavedeme do výpočtu skóre časovou dynamiku, skóre znevýhodněno v závislosti na rozdílu mezi rokem vydání publikace a momentálním rokem.

$$sim_{u,p} = f_y(d_y) \cdot P(a_u, a_p) \tag{13}$$

kde

- $sim_{u,p}$ je podobnost mezi autorem a publikací s přidáním časové dynamiky
- f_y je funkce časové dynamiky nabývající hodnoty $\in [0; 1]$
- a jsou porovnávané atributy



Obrázek 10: Grafy funkcí časových dynamik

- P je podobnost mezi atributy
- d_y je rozdíl mezi momentálním rokem a rokem vydání publikace porovnávaného atributu a_p

Volba funkce časové dynamiky je závislá na konkrétním případě. Můžeme např chtít, aby pro obory medicíny nebo fyziky nebyly staré publikace nijak znevýhodňovány, pro počítačové vědy zase opak. Funkce f_y zadáme jako

- *Funkci konstantní* $f_y = 1$ pro případy neznevýhodňující stáří publikací
- *Funkci lineární* $f_y = -0,05d_y + 1$ pro případy znevýhodňující starší publikací rovnoměrně
- *Funkci logaritmickou* $f_y = log_{20}(-d_y+20)$ pro případy znevýhodňující starší publikací nelineárně

Rozdíly mezi funkcemi jsou zobrazeny v grafu na obrázku 10. Logaritmická funkce více reflektuje zapomínání v čase u člověka v porovnání s funkcí lineární. Přidáním časové dynamiky vzniknou lehce modifikované verze Jaccardova indexu (viz vzorec 14) a Diceova koeficientu (viz vzorec 15).

$$J_t(A,B)\frac{\sum_{i=1}^{|A\cap B|} f_y(d_{y,i})\cdot a_i}{|A\cup B|} \tag{14}$$

$$d_t(A,B)\frac{2\sum_{i=1}^{|A\cap B|} f_y(d_{y,i})\cdot a_)}{|A|\cup|B|} \tag{15}$$

kde

- A a B jsou porovnávané množiny klíčových slov

- a je klíčové slovo $\in A \cap B$
- $d_{y,i}$ je rozdíl momentálního času a roku vydání autorovy publikace atributu a_i
- f_y je výše zmíněná funkce časové dynamiky

Protože lidská řeč je velmi rozmanitá a jedna myšlenka se dá vyjádřit více způsoby, přidáme do algoritmu hledání a porovnávání také na základě synonym. Zvolíme vhodný slovník (v našem případe WordNet) a pro jednotlivá klíčová slova rozšíříme na seznamy možných kombinací jejich synonym. Například pro *'test software'* vytvoříme seznam *['test software', 'test application', 'check software', 'check application', ...]*. Seznamy ještě setřídíme podle abecedy pro snadnější porovnávání, rozhodnutí zda klíčová slova jsou stejná se vyjádří vztahem:

$$decision = \begin{cases} k_1 = k_2 & \text{jestli} |K_1 \cap K_2| > 0 \\ k_1 \neq k_2 & \text{jinak} \end{cases} \tag{16}$$

kde k_1 a k_2 jsou původní klíčová slova a K_1 a K_2 jsou množiny klíčových slov rozšířených o synonyma.

Navržený postup není samozřejmě omezen pouze na doporučování publikací uživatelům. Algoritmus se dá aplikovat také na hledání vhodných hodnotitelů pro publikace (např. pro přihlášené články na konferenci), můžeme stejným způsobem dohledat podobné publikace nebo autory publikující v podobné oblasti výzkumu. Jen v těchto případech nemá příliš smysl postihovat starší publikace, je proto vhodné vynechat rozměr časové dynamiky.

5.6 Použité technologie

Pro ověření funkčnosti a správnosti navržených algoritmů (viz předchozí kapitola 5) byla vytvořena webová aplikace v jazyce Python. Z důvodů použitých knihoven a záměrem dostupnosti systému na webu je primárně aplikace vyvíjena pro operační systém Linux, ale měla by být spustitelná i na dalších operačních systémech (Windows, MacOS X). Následující část popisuje implementaci vybraných dílčích částí práce.

Python

Python[12] je objektově orientovaný dynamický open source programovací jazyk vydaný v roce 1991 Guido va Rossumem. Významnou vlastností tohoto jazyka je produktivnost z hlediska rychlosti psaní programů a značné úspory napsaného kódu. Popularita Pythonu dlouhodobě mírně roste a využití nachází jak při tvorbě webových a desktopových aplikací, tak ve vědecké sféře. V diplomové práci je použit Python ve verzi 3.3.

[12] http://www.python.org/

Django

Django[13] je robustní open source webový framework napsaný v jazyce Python. Implementuje architekturu MVC a řadí se mezi nejrozšířenější frameworky v ekosystému Pythonu vůbec. Díky rozsáhlému počtu pluginů třetích stran a návrhu architektury, je vývoj webových aplikací rychlý.

NLTK a Textblob

Knihovna NLTK[14] je přední platformou pro vytváření aplikací v Pythonu pracující s daty lidského jazyka (tzv. NLP[24]). Poskytuje snadné rozhraní k přístupu více než 50 korpusům a lexikálním prostředkům jako je např. WordNet. Spolu s balíkem zpracování textu implementuje také knihovny pro klasifikaci, tokenizaci, hledání slovních druhů a analýzy sémantiky textu.

Textblob[15] je knihovna rozšiřující NLTK a přidávající některé další funkce používané při analýze textu. Při implementaci je z této knihovny použit Naivní Bayesův klasifikátor, hledání synonym a frází podstatných jmen.

SeleniumHQ

Selenium[16] je rozhraní umožňující automatizovat práci s webovým prohlížečem sadou předepsaných příkazů. Prvotní myšlenkou vzniku bylo zautomatizování webových aplikací za účelem jejich testování. Využít se dá ale také pro dolování dat z webu v případech, kdy běžné zasílání HTTP požadavků není možné, například kvůli velké závislosti obsahu na technologiích, jako jsou Javascript, Flash apod.

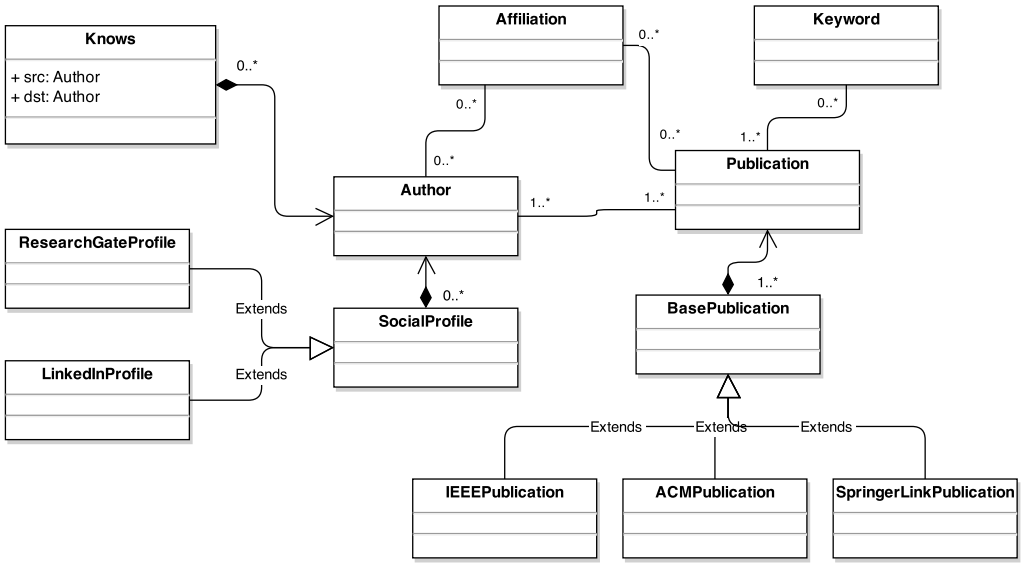
AngularJS

AngualarJ[17] je Javascriptová knihovna vytvořena pro snadnější vývoj webových aplikací. Využívá architektury MVC, veškerá prezentace dat probíhá na straně klienta, se serverem dochází ke komunikaci pomocí REST, RPC nebo SOAP služeb. V implentované aplikaci je zvolen přístup na RPC a prezentaci dat v JSON formátu.

Twitter Bootstrap 3

Twitter Bootstrap[18] je CSS framework vytvořen pro potřeby vývoje responzivního a tzv. *mobile first* webových projektů. V této diplomové práci je využit pro prezentaci dat uživateli a vybrán byl hlavně kvůli jeho rozšířitelnosti, množství dostupných rozšíření a responzivního designu.

[13] https://www.djangoproject.org/
[14] http://www.nltk.org/
[15] http://textblob.readthedocs.org/en/dev/
[16] https://code.google.com/p/selenium/
[17] https://angularjs.org/
[18] http://getbootstrap.com/



Obrázek 11: Třídní diagram aplikace

Relační databáze

Pro úložiště nalezených a zpracovaných dat byla vybrána SQL databáze SQLite3, jenž je vhodnou volbou při provozu aplikace na klasickém osobním počítači. Software je navržen tak, že je snadné změnit databázový engine na téměř jakoukoliv jinou SQL databázi. Podporovány jsou konkrétně MySQL, PostgreSQL, Microsoft SQL Server Oracle nebo vhodné pro provoz aplikace na serveru.

Mapování relačních dat tabulek na objekty probíhá pomocí integrovaného ORM v Django frameworku. Třídní diagram navržené databáze je znázorněn na obrázku 11.

6 Výsledky experimentu

Výsledek této diplomové práce, tedy navržený algoritmus, byl ověřen na experimentu, který měl dokázat jeho funkčnost a zjistit úspěšnost. Testování bylo rozděleno na dvě části. V první fázi se ověřovalo hledání identit uživatele v digitálních knihovnách a na sociálních sítích s 30 náhodnými autory. Ve fázi druhé se ověřovalo doporučování publikací autorům dle porovnání reálných dat ze dvou konferencí.

6.1 Experiment hledání identit

Cílem tohoto testování bylo ověřit a prokázat funkčnost správného nalezení identity uživatele jak mezi digitálními knihovnami, tak napříč sociálními sítěmi.

Pro test v rámci knihoven bylo testováno hledání 180 autorů náhodných národností. Výsledky jsou uvedeny v tabulce 2. Nejdříve bylo provedeno testování se seskupováním publikací podle spoluautorů a institucí. Celkem 119 autorů bylo rozpoznáno správně (sloupec „S“), 2 autorům byly přiřazeny publikace jiných autorů (sloupec „PJA“) a žádný nebyl špatně sloučen s jiným autorem (sloupec „NP“). Poměrně nízká hodnota neúspěšností je kompenzována vysokým počtem nepřidělených publikací (sloupec „NS“), respektive pro publikaci bylo vytvořeno více individuálních identit. Tento případ nastával v situacích, kdy autoři publikovali sami, nebo zřídka s zvedenými spoluautory. Pokud bychom brali v potaz špatné výsledky, úspěšnost tohoto přístupu je 98,3 % a 1,6 % chybovost. V dalším testovacím případě se k předchozí metodě seskupení přidalo krok s hledáním podobnosti klíčových slov. Skupiny obsahující jednu publikaci byly porovnávány s ostatními a případně shody k nim přiřazeny. Výsledkem je 169 správně rozpoznaných publikací, 8 bylo přiřazeno k jiných autorů a 3 uživatelé byli sloučeni nesprávně s jinými. Úspěšnost tohoto testu byla 93,3 %, o cca 5 % méně než v předchozím případě, ale s větším množstvím přiřazených publikací.

K testování mezi knihovnami a sociálními sítěmi bylo vybráno 30 náhodných uživatelů, u kterých byla ručně ověřena přítomnost na všech implementovaných službách, tedy na sítích LinkedIn, ResearchGate, ACM DL, IEEExplore, SpringerLink. Výsledky jsou znázorněny v tabulce 3. Pro 30 hledaných osob bylo správně nalezeno 23 profilů na síti LinkedIn a 27 profilů na síti ResearchGate. Lepší skóre posledně jmenovaného je pravděpodobně způsobeno velkou podobností služby s digitálními knihovnami, kde princip je založen také na hledání podobnosti dle publikací. Koncept na základě porovnávání kolegů a publikací se tedy ukazuje být správným. Po manuálním ověření nenalezených autorů bylo zjištěno, že jejich profily obsahují velmi málo informací nutných ke správnému rozhodnutí o korektnosti identity. Ve většině případů obsahovaly pouze jméno a pár spojení s jinými uživateli, dva profily nebyly veřejně přístupné.

6.2 Experiment doporučování publikací

Cílem tohoto pokusu je prokázat a zjistit schopnost doporučení správných publikací autorům na základě jejich předchozí publikační činnosti a to způsobem přidělování článků

Metoda seskupení	S	PJA	NP	NS
Spoluautoři	118	3	0	59
Klíčová slova + spoluautoři	168	9	3	0

Tabulka 2: Výsledky experimentu hledání publikací

Počet autorů	LinkedIn	ResearchGate	LinkedIn %	ResearchGate %
30	23	27	73 %	90 %

Tabulka 3: Výsledky experimentu hledáni identit

přihlášených na konference vhodným hodnotitelům. Testovací scénář je založen na reálných datech dvou konferencí, jedné menší lokální čítající 19 členů - posuzovatelů a 19 přihlášených článků. Druhá globální konference se skládá z 90 členů a 110 přihlášených článků. Všechna data byla zpracována anonymně a použita pouze pro potřeby tohoto experimentu. Skóre (shodnost) publikací a autorů bylo počítáno podle Jaccardova indexu a Diceova koeficientu (viz kapitola ...).

Ověření správnosti má dva výsledky. V prvním se ručně kontroluje, zda přidělené autoři byli správně přiděleni, tedy jejich publikační činnost je podobná a jsou vhodnými kandidáty pro hodnocení článku. Druhý výsledek určuje shodu s reálnými přiřazeními hodnotitelů k publikacím, tedy jestli nalezení hodnotitelé jsou ti, kteří skutečně danou publikaci hodnotili. Myšlenka ověřování výsledků je následující: Pokud alespoň čtyři z prvních pěti nalezených hodnotitelů seřazených podle skóre sestupně mohou hodnotit publikaci, považuje se výsledek za správný. V případě ověření s reálnými daty se za úspěšné považuje, pokud mezi prvními pěti nalezenými hodnotiteli seřazenými sestupně je alespoň jeden, který byl mezi skutečnými hodnotiteli článku na konferenci. Poslední kontrola je uvedena především pro představu, jak moc se algoritmus shoduje s lidským přiřazováním, a není skutečně vypovídající. Výsledky experimentu jsou znázorněny v tabulce 4.

První sloupec obsahuje počet přiřazovaných článků pro všechny hodnotitele, druhý

	Článků	Manuální ověření	Ověření s daty
dice	19	16	13
jaccard	19	16	13
dice	110	79	72
jaccard	110	79	71
dice(synonyma)	19	17	14
jaccard(synonyma)	19	17	14
dice(synonyma)	110	84	75
jaccard(synonyma)	110	86	73

Tabulka 4: Výsledky experimentu doporučení publikací

sloupec popisuje počet správně přiřazených publikací s manuálním ověřením, zda nalezení hodnotitelé jsou potencionálně správní a třetí - poslední sloupec jsou hodnoty správně přiřazených autorů v porovnání se skutečnými daty z konferencí.

V první polovině tabulky jsou výsledky pro Diceův koeficient a Jaccardův index s hledáním podobnosti bez zahrnutí synonym. Jak lze vidět, úspěšnost se pohybuje okolo 84 % u menší konference a 72 % v případě větší. Lepší výsledky dosažení u lokální konference jsou pravděpodobně menším poměrem množstvím publikací na autora (okolo 15) a tím větší přesnosti odhadu klíčových slov. U větší konference připadalo na jednoho hodnotitele cca 3x více publikací, došlo tak k většímu zkreslení velkým množstvím nalezených dodatečných slov. Dobrým momentem bylo, že pro jednoho účastníka menší konference nebyl nalezen žádný přihlášený článek vhodný k ohodnocení. To bylo způsobeno faktem, že daná osoba nepublikovala v oblasti specializace konference, výsledek byl tedy správný.

V další části experimentu byl vylepšen algoritmus přidáním synonym ze slovníku WordNet ke klíčovým slovům, včetně dodatečně přidaných. Výsledky jsou popsány v druhé polovině tabulky 4. Skóre se mírně zvýšilo, u lokální konference na 89 % a 77 % pro konferenci globální. Bohužel se ale také rozdíl skóre mezi publikacemi a jejich hodnotiteli zmenšil. Někteří členové získali lepší skóre než v prvním scénáři bez synonym, tedy získali vyšší prioritu pro hodnocení publikace nad jinými, kteří by subjektivně měli hodnotit spíše. Tento přístup s využitím synonym by mohl být využíván v případech, kdy obor konference je velmi rozmanitý a přihlášené články také. Mělo tímto být zajištěno, že se najde vhodnější kandidát k hodnocení.

7 Závěr

Cílem diplomové práce bylo navrhnout a implementovat algoritmus pro hledání identit uživatelů v digitálních knihovnách a na sociálních sítích a algoritmus doporučující články uživatelům na základě jejich předchozí publikační činnosti. Oba algoritmy měly být vytvořeny za účelem usnadnění práce s vyhledáváním identit uživatelů na sociálních sítích, doporučovací algoritmus pak za účelem přiřazování vhodných článků vybraným hodnotitelům při pořádání konferencí. Výsledky a vhodnost návrhu byly ověřeny na testovacích případech založených na skutečných datech ze dvou konferencí.

Nejdříve bylo nutné seznámit se s poznatky v oblasti podobného výzkumu a možnostmi řešení. Analyzoval jsem data zveřejňovaná na sociálních sítích a digitálních knihovnách a podle nich dále hledal možné cesty a východiska. Na základě mnoha testů a experimentů, např. s hledáním ontologií, domény na základě slovníků, atd., ještě před samotným návrhem jsem se rozhodl pro přístup založený na porovnávání předem vytipovaných atributů na jednotlivých profilech s různým přiřazením vah. Potřebné informace k určení domény uživatele jsem se rozhodl získat analýzou abstraktů a porovnáváním atributů v modelech vektorového prostoru. Při práci bylo potřeba vyřešit problémy hlavně se stahováním dat z webových služeb. Mnoho z nich omezuje přístupy ke svým datům různými limity, bylo tedy nezbytné nasimulovat lidské chování při stahování dat. Během návrhu a implementace se bohužel měnily podmínky užívání služeb a také došlo k zamezení přístupu k některým informacím uživatele. Tyto problémy jsem musel řešit v průběhu času. Nelze tedy zaručit, že popsaný přístup a implementace budou funkční stále.

Ve fázi testování se ukázalo, že navržené algoritmy hledají relevantní výsledky o profilech uživatelů a doporučují vhodné publikace, hlavně pro autory mající přiřazeno menší množství publikací. S rostoucím počtem klíčových slov již docházelo k většímu zkreslování o doméně autora.

S návrhem algoritmů se ukázala celá řada dalších možných způsobů a cest ke zlepšení. Přidáním dodatečných vstupních parametrů při vyhledávání autora (např. spoluautor, klíčové slovo domény, instituce) by se mohl vyloučit krok manuální verifikace nalezených údajů a celý proces by se v podstatě dále plně zautomatizoval. Vhodné by bylo vylepšit systém hledání synonym a zařadit také metodu jazykového modelování pro zlepšení podobnosti mezi publikacemi.

Dalším směrem této práce může být komplexnější popis autorova profilu. Ten může sloužit například pro další ověření porušení copyrightu článku s porovnáním jeho předchozí publikační činnosti. Doména autora se nemusí získávat jen z publikací, ale mohou být obsaženy některé informace ze sociálních sítí. A ještě lépe, mohou se do ní zahrnout také výsledky fulltextového vyhledávání na webu. Vzdáleným cílem by mohlo být nalezení jednotné identity uživatele v rámci celého dostupného Internetu. Z trochu jiné oblasti by popsané postupy mohly být základem pro hledání důležitých částí v textu a vztahů mezi nimi. Aplikací by mohlo býti nalezení atributů zadaných produktů na základě analýzy stránek z výsledků fulltextu.

Na základě této diplomové práce vznikl odborný článek *Social network and Digital Library User's Identification*, který bude odeslán na konferenci ACM HyperText 2014 konající se v Chile. Článek je přiložen k této diplomové práci.

8 Reference

[1] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[2] H. Kautz, B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, March 1997.

[3] J. Vosecky, Dan Hong, and V.Y. Shen. User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 360–365, July 2009.

[4] E. Raad, R. Chbeir, and A. Dipanda. User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, pages 297–304, Sept 2010.

[5] G. Hurtado Martín, S. Schockaert, Ch. Cornelis, and H. Naessens. Finding similar research papers using language models. In *2nd workshop on semantic personalized information management : retrieval and recommendation, Proceedings*, pages 106–113. University College Ghent, 2011.

[6] T. Segaran. *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly, Beijing, 2007.

[7] A. H. Lashkari, F. Mahdavi, and V. Ghomi. A boolean model in information retrieval for search engines. In *Information Management and Engineering, 2009. ICIME '09. International Conference on*, pages 385–389, April 2009.

[8] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[9] G. L. Giller. The statistical properties of random bitstreams and the sampling distribution of cosine similarity. *Available at SSRN 2167044*, 2012.

[10] P. Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.

[11] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons. *Biol. Skr.*, 5:1–34, 1948.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[13] A. Lukasová and J. Šarmanová. *Metody shlukové analýzy*. SNTL, 1985.

[14] T. H. Cormen, C. Stein, R. L. Rivest, and Ch. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[15] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, New York, NY, USA, 1997.

[16] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

[17] F. J. Damerau. A technique for computer detection and correction of spelling errors. *Commun. ACM*, 7(3):171–176, March 1964.

[18] W. E. Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359, 1990.

[19] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, June 1989.

[20] M. F. Porter. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.

[21] P. Chmelař, D. Hellebrand, M. Hrušecký, and V. Bartík. Nalezení slovních kořenů v češtině. In *Znalosti 2011: Sborník příspěvků 10. ročníku konference*, pages 66–77. VŠB-Technical University of Ostrava, 2011.

[22] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION*, pages 41–48. AAAI Press, 1998.

[23] M. Bayes and M. Price. An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to John Canton, M. A. and F. R. S. *Philosophical Transactions (1683-1775)*, 1763.

[24] S. Bird, E. Klein, and E. Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

A Obsah přiloženého CD a článek HyperText 2014

```
DiplomovaPrace
├── aplikace
│   ├── diplomka
│   ├── core
│   ├── web
│   └── README
├── text_dp.pdf
└── clanek_hypertext_2014.pdf
```

Social Network and Digital Library User's Identification

Petr Saloun[1], Adam Ondrejka[1], Ivan Zelinka [1]
[1]*VSB - Technical University of Ostrava*
petr.saloun@vsb.cz, adam.ondrejka.st@vsb.cz, ivan.zelinka@vsb.cz

Keywords: digital library, identify user, social media, papers recommendation, information retrieval, natural language processing

Abstract: Two main goals of this work are design algorithm to find identities of users through social networks and digitial libraries and algorithm to recommend publications to reviewers on conferences. In this paper we focus on estimating identity and research area by analysis of publicly available metadata about authors and their publications and informations published on social networks. By right estimation of the domain in which the author is publishes and on properly selected keywords and co-authors we can quite efficiently alocate the identity of users and recommend other appropriate publications or areas of research. The result is design and experimental verification of algorithm, which are part of this paper.

1 INTRODUCTION

The boom of social networking has caused the problem in the form of a fragmentation of accounts / identities per user. Most of these networks creates a new restricted user accounts, and are difficult to work with the user as a single entity. There are some projects seeking to identify through a single account (eg OpenID), but the implementation of operators avoid social platforms. At the same time operators allow identification through an account in selected social networks.

The aim of this work is to look for researchers across selected social networks on the basis of their publications and interests published on profile pages eventually. The second goal is to create recommendation algorithm and system based on collected data about researches. User enters basic information about the article and algorithm automatically finds the most appropriate researchers by their domain of research.

In ideal world each man has his unique name or ID and it will be absolutely sure who is he. In the real world this is unfortunately not and the authors may have similar or identical names, whether similar or completely different areas in which the audience. Image situation you are trying to find author (eg someone you met on conference). When you type his name in digital library these scenarios could occur:

1. Author has unusual name or even unique one. No one else with this name doesn't publish, we find him immediately.

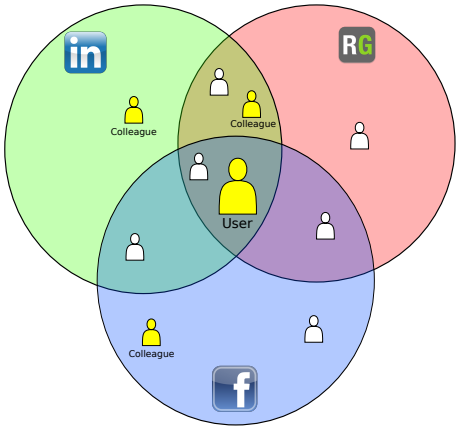


Figure 1: User's colleagues on more social networks.

2. Author has usual name. We find a lot of people with this name, but luckily they research in different areas, don't work on same institute and have others colleges. So we can guess the person we was looking for.
3. Author has very usual name. We find a lot of people and unfortunately they are in same or similar area of research, they work on same institute or have same colleges.

As of the above scenarios third point is very problematic and difficult to assess even with manual methods. Furthermore, we focus primarily on the first two scenarios.

Finding similar publications by typical access

with comparing each word is not suitable for our case. We have a small amount of information (title and abstract) so only by comparing words we would received misrepresentation. It's necessary to try understand simple semantic of text and compare synonyms of words too. For these purpose WordNet (Fellbaum, 1998) dictionary will be used.

2 STATE OF THE ART

J. Vosecky in his article *User identification accross multiple social networks* (Vosecky et al., 2009) identifies users across two social networks by similarity of user profiles. A similar approach was chosen by E.Raad in his algorithm described in *User Profile Matching in Social Networks* (Smith, 2010). Raad assigns different weights and scores to selected attributes on user profiles and unify them under a specific identity on the basis on final score.

For subtask of recommending publications to researches was described many algorithms and articles. This is similar problem, which is today largely solved in business and e-commerce. Companies want to encourage customers tu purchase other goods by shopping habits and one of them is just finding similar users and offering goods to chosen customer has not purchased yet and other did. In book *Programming Collective intelligence* explains Tony Segaran that in such a case the data are converted to vector space and then the score is calculated based on distance between them. This score (depends on method to calculate) expresses similarity.

3 SOCIAL NETWORKS AND DIGITAL LIBRARIES

From the digital libraries were chosen for the experimental purpose IEEExplorer [1], ACM Digital Library [2] and SpringerLink [3]. IEEE and SpringerLink libraries have public API, but allows access to different data. It is therefore necessary to get additional attributes directly from libraries pages.

In this work, the researchers are finding in social networks LinkedIn [4] and Researchgate [5]. These networks have good API and allow users to show enough information for next research. LinkedIn is site primarily focused on jobs and coworkers, but offers a lot of useful data and informations to assess. Researchgate is relatively new social network intended for scientist. His users publish their publications, connect with coworkers and discuss the topics in their research areas.

Unfortunately well known sites like Facebook or Twitter can not be included in this experiment. In last time they significantly restrict access to public data of users and it is almost impossible to obtain data needed for the experiment now.

[1] http://ieeexplorer.ieee.org

[2] http://dl.acm.org

[3] http://www.springerlink.com

[4] http://linkedin.com

[5] http://www.researchgate.com

4 ALGORITHM DESIGN

4.1 Publication analysis

Since the foundation of the work is finding and identifying areas of users research based on their publications, the next chapter is devoted to publication analysis. If output of this process will be adequately fit and proper, we can expect very good result in identify right users.

Firstly we need obtain meta-data of publication from digital libraries. It's important to web crawler will get these informations: *Title* as identification and for keywords analysis,*Abstract* to guess article domain and keywords analysis, *Keywords*, *Authors* to find out possible friends on social networks (we don't want editors) *Affiliates*, *Year of publication* and *References*.

Problem is that every digital library returns data parsed differently. Some sites provide author in format with full name, other sites show authors with full last name and first letter of first name or middle name. This complicates situation little bit, we have to guess if co-authors are really who we think they are.

Because provided keywords are often very little and in many cases completely missing, we need to analyse abstract and process it to additional words. For this we use NLTK library in Python for processing natural language [6]. Algorithm is detailed descibed in chapter 4.5.

Affiliates are very useful to correct assignment of authors and their co-workers. Unfortunately affiliates are label on each digital library differently even in same library. In this case is good to compare affiliates by text similarity algorithm. Partial ratio using sequence matcher published in *Pattern Matching: The Gestalt Approach* (Ratcliff and Metzener, 1988) compare strings sufficiently for our experiment.

[6] http://www.nltk.org/

4.2 Finding authors on digital libraries

As shown in algorithm 1 user enters the name of desired user. Then search requests for all digital libraries is executed and waiting for results of publications. Each publication is then grouped by defined criteria. Firstly we eliminate same articles occurring on more libraries. Then we group them by affiliations using the string similarity algorithm described above and next step is analysis publications described above too. Then author is queried in our database and we trying to link him if already exists.

```
Data: Author's first name and last name
Result: User's identities
firstName, lastName ← {user raw input};
for searcher in DigitalLibrariesSearchers do
    publications ←SearchAuthor(firstName,
    lastName);
end
GroupByPublication(publications);
GroupByAffiliates(publications); for
publication in publications do
    AnalyzePublication(publication);
end
if author in database then
    {compare find publications with
    publications in database and remove
    existing };
    {compare authors };
end
AddNewAuthors();
AddNewPublications();
```

Algorithm 1: Finding authors on digital libraries

After removing existing authors and publications in database the new ones are added. This is repeat for every found co-authors and references in publication with input of found authors so results will be more accurate due these restrictions. Uncertain authors which identity can not be definitely determined (e.g. there are many publications with one author publishing independently) could be specified by using a similarity search between them based on keywords analyse.

4.3 Finding authors on LinkedIn

Each social network has different public data of its users so individual access is required. On LinkedIn profile are particularly interesting information about connection with other users, jobs, experiences and publications. If user had filled in at least three of these data it can determine its identity with very high accuracy. Otherwise we will have to guess whether it really is about. Entry conditions for this algorithm are processed information about the author of digital libraries and its identity in the database.

```
Data: Author's first name and last name
Result: User's identity on LinkedIn
firstName, lastName ← {user raw input};
users ← FindLinkedInUser();
for user in users do
    find user's connections, jobs, experiences
    and publications;
end
find user in database without linkedin
connection;
for each found user do
    compare experiences, connections with
    co-authors, publications and domain if
    exist;
    calculate score;
end
get user with highest score;
if user with highest score >minimum score then
    we find user;
end
```

Algorithm 2: Finding authors on LinkedIn

Author is searched by LinkedIn API. Next step is analyse each found user and get info about his experiences, hobs, publications and connections. If we previously collect enough quantity of data we can guess right user with very high probability. There is a big chance user will have connection with people who are his co-workers and co-authors too. Also jobs position could be affiliation listed in his publication. Although string comparing is needed again but with time the exact links between co-authors and LinkedIn increase and guess accuracy too of course.
If user had filled in info about his skills and experiences we try to guess domain of research. This should help with later decision. Each user in database without LinkedIn connection is compare with found result and calculate score for this pair. Highest score is then compared with required minimum and if is high enough, we find our user.

4.4 Finding authors on ResearchGate

ResearchGate can be described as mix of digital library and LinkedIn so identify user on this site is very similar to access used on both mentioned above. Most of users have filled in topics of their interest what should be mean as areas of research. Users also follow other users. In these lists we can find co-authors and jobs are big candidates for affiliates.

```
Data: Author's first name and last name
Result: User's identity on ResearchGate
firstName, lastName ← {user raw input};
users ← FindResearchgateUsers();
for user in users do
    find user's connections, jobs, experiences
    and publications;
end
find user in database without researchgate
connection;
for each found user do
    compare experiences, connections with
    co-authors, publications and domain if
    exist;
    calculate score;
end
get user with highest score;
if user with highest score >minimum score then
    we find user;
end
```

Algorithm 3: Finding authors on ResearchGate

4.5 Recommendation of publications

By analysing of the whole publication we should relatively precisely estimate the domain of publication and we can say about author whether he would meet another article or not. This is very helpful for expanding user knowledge and area of his research. Unfortunately we have access only to short text in the form of abstract and prior testing has shown that guess of domain was not good enough for the final decision. But it should be eventually used to assess whether the article is totally out of area or not. So we have decided to choose a slightly different path by analysing abstract, picking the right keywords. Algorithm focus on comparing keyword in lemma form and calculate similarity between synonyms available by WordNet.

Firstly we detect abstract language and translate it to English if is not. This is done by Google Translate API which is very accurate and adequate. Next we try discover important keywords in abstract text by natural language processing by NLTK library. At the beginning we make tokens from text and find part-of-speech tag which are analysed. These will serve in next step when we will doing named entity recognition and keep all words detected as person. On the other hand words marked as location, date, time and facility are removed. The reason to keep only person named and geo-political entities is that very important keywords like technologies names has tendency to be detected as these as we find out during our analysis and the other group was in most cases meaningless.

```
Data: Abstract
Result: Additional keywords
lang ← DetectLanguage(abstract); if lang is not
'en' then
    TranslateAbstract(abstract);
end
tokens ← tokenize(abstract);
tags ← PosTagger(tokens);
nerTokens ← NeChunker(tags);
for each ner in nerTokens do
    if ner is not person then
        remove from abstract
    end
end
for each tag in tags do
    if tag is negation then
        label tag as 'NEG' ;
    end
end
NPS ← get noun and negation noun phrases;
for each phrase in NPS do
    if phrase is noun phrase then
        phrase ← Lemmatize(phrase);
        phrase ← RemoveStopWords(phrase);
        if ClassifyPhrase(phrase) is keyword
        then
            add phrase to additional keywords;
        end
    end
end
return additional keywords;
```

Algorithm 4: Extract additional keywords from abstract

Abstract is without distorting words now, but still should contains negations. All these words we mark with label "NEG" by looking up in predefined dictionary and then we find noun and negation noun phrases by regular expressions. This is not the best approach, but showed be good enough. Some important keywords should be removed but it is better, then include one which can distort the results. Plus negations are not so frequent in abstracts.

```
NP: {<JJ|NN.*>\}
NP-NEG: {<*NEG><DT|JJ|NN.*>+}
```

Next we lemmatize positive noun-phrases and convert it into singular. After each phrase is deprived of stopwords and classify by Naive Bayes classifier (Devroye et al., 1996). This was preceded by creation of learning set of 6000 marked keywords which is as input to classifier used in our analysis. Now we have two groups sets and sure we keep only noun phrases because these are our new additional keywords.

Comparing authors and publications will be com-

puted in vector space, but before we have to find synonyms for each keywords by Wordnet library which provides satisfactory dictionary of two hundreds synonyms words and their similarity. It prevents false comparison only by text comparing. Before this process each keyword will be stemmed. This is happening because of text string comparing, e.g. words *design*, *designing*, *designed* are all in lemma form aren't good to compare. By extract stems, each previous words give result *design* which is better. All composed stemmed keywords are sorted alphabetically at preparation for better comparison.

Finding text similarity should be done by more methods, e.g language modelling (Martin et al., 2011), but we primarily focus on similarity in vector space.

5 Experiment

To build a test collection we use real data from two conference, one local and the other global area. Local conference (Czech Republic) consists of 19 PC members and 19 registered papers with all bids. Global conference data has 90 PC members and reviewers and 110 registered papers.

Experiment has two goals. First is try find identities of registered papers authors on social networks LinkedIn and ResearchGate. Secondly we will try assign papers to rating to right PC members based on their own publications. This result will be compared with real bids. To find similarity between publications in vector space we will use these distances:

- **Jaccard index** (Jaccard, 1912) - used for comparing similarity and diversity of sample sets, equation 2
- **Dice coefficient** (Dice, 1945) - equation 2

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

$$d(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

Where A means sets of publication 1 keywords and B means sets od publication 2 keywords

The ground of truth for our experiments is based on manual verification of algorithm output. For finding authors on social networks were randomly chosen 30 people and manually found their profiles, then the algorithm output will be compared with prepared results. For publication-reviewers assignment problem will be algorithm output compared with data from conference and again with manual verification. This is necessary because not matching results don't have to mean they are wrong.

5.1 Results

Search user identities on digital libraries have been tested for 180 researchers by downloading and analysing about 5770 publications. 169 users was identify correctly, 8 users had assigned wrong publications and 3 users were merged with other users inaccurately. Success ration was about 93.9 %.

Table 1 summarize results of the social networks experiment. For 30 searched person were found 23 correct profiles on LinkedIn site and 27 profiles on ResearchGate. Better score of ResearchGate is probably caused because of the similarity this social network with digital libraries so concept of assessment profiles based on comparing colleges and publications is right. After manual check unfound users was determined their profiles don't content necessary information to make a good decision. In all cases there were filled only users' names and a few connections.

Table 1: Experiment 1 results

Count	LI	RG	LI %	RG %
30	23	27	73 %	90 %

Table 2 shows results of experiment no.2. First column describes number of analysed publication for the all evaluators, second column means count of successfully assigned papers checked manually and in third column are values of papers count compared with real bids. The thought of verifications is following: If at least four in top five evaluators sorted descendent by score should judge publication the it is considered as success for manual checking. Right result for data based verification is such that contains at least one evaluator presented in real bids in top five matches. This last score is shown for idea how much is algorithm coinciding with human assignment and not really meaningful.

First row group of table summarize output for comparing papers and reviewers by Jaccard's and Dice's index without synonyms. As we can see success assignment is around *84 %* in small conference and *72 %* in bigger one. Better result in first case is probably caused by smaller quantities of reviewer's publications (about 15 per reviewer), at second case amount was three times greater per user and that may distortion of information. A good moment was that the researcher evaluating the smaller conferences were not assigned to any article, because in fact it is his area of research is other than the conference specialization.

Table 2: Experiment 2 results

	Papers	Manual ver.	Data ver.
dice	19	16	13
jaccard	19	16	13
dice	110	79	72
jaccard	110	79	71
dice (syn)	19	17	14
jaccard (syn)	19	17	14
dice (syn)	110	84	75
jaccard (syn)	110	86	73

In next part of experiment we improve algorithm by adding keywords synonyms available by WordNet dictionary. Results are described in second group and improvement is noticeable but not significant. Score increase little bit, to *89%* for small conference and *77%* for bigger one, but also differences between various publications decreased. Some reviewers had better score than in first scenario which means they got priority over others who had evaluated publications rather (but this is very subjective criteria). This approach with using synonyms should be used in scenarios where reviewers areas of research are more diverse and logged papers too. It should be more accurate to find right evaluators.

6 CONCLUSIONS

The main goals of our work were to design and implement two experimental algorithms, one to identify user on digital libraries and social networks, second to recommend appropriate publications to researchers. We have chosen approach of analysis public meta-data of researchers and guess their possible research area.

For experiment were chosen digital libraries ACM, IEEExplorer and SpringerLink, social networks LinkedIn and ResearchGate. Similarity between users and publication was based on vector space model as one of the possible approach of many.

As our experiment results show, algorithm to searching user on digital libraries was successful in 94% of searching. Finding them on social networks by analysed publication meta-data depended on each site individually. More successful (90%) was ResearchGate web because of its similarity with digital library. LinkedIn success ratio was 76%.

Recommendation algorithm has showed that best results are reached with lower numbers of assigned publications. For 19 articles and about 15 publications per user algorithm correctly recommend in 84% cases, for 110 articles and 90 reviewers it was about 72%. Adding synonyms comparing improve success to 89% for small and 77% to bigger conference.

For the improvement of algorithm domain guess, ontology and better natural language processing should be implement and also should be reflected year of publishing because of penalty to too old research areas. The next way should be use different scoring computation and include information of social networks too. Right join of user accounts though web sites could carry and help to create a real area of the user and his areas of interest and research.

REFERENCES

Devroye, L., Gyorfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer.

Dice, L. (1945). . measures of the amount of ecologic association between species. *Ecology*, (26):297–302.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. *THE NEW PHYTOLOGIST*, 11(2):37–50.

Martin, G., Schockaert, S., Cornelis, C., and Naessens, H. (2011). Finding similar research papers using language models. *2nd workshop on semantic personalized information management : retrieval and recommendation, Proceedings*, pages 106–113.

Ratcliff, J. W. and Metzener, D. E. (1988). Pattern matching: The gestalt approach. 13(7):46, 47, 59–51, 68–72.

Smith, J. (2010). User profile matching in social networks. In *Network-Based Information Systems (NBiS), 2010 13th International Conference on*, Takayama. The publishing company.

Vosecky, J., Hong, D., and Shen, V. (2009). User identification across multiple social networks. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 360–365.